July/August 1982 Volume 2, Number 4 $2.95 (USA) £1.50(UK)

The magazine for Sinclair users

SYNC®

Use the DEFine function To Construct 3-D Plots

Sinclair ZX Spectrum: An In-depth Review

"To Explore New frontiers..." Six Games of Outer Space

Hardware: More Memory, Power Filtering, Ear Input

SINCLAIR APPLICATIONS PRESENTS ROM CARTRIDGES FOR THE ZX-81.*

ROM PACKS

- Eliminate the flustration of loading cassettes
- Enable long programs without long load times or extra memory
- Can't be erased, never fail to load

CHARACTER CARTRIDGE

- 3 on ROM character sets
- Program selectable
- Upper/lower case character set
- Graphics character set
- Math/electronics character set
- None of the above take up any user RAM
- With 512 bytes of user RAM the ROM PACK can build a Spanish or German character set with Ö and ¿ type characters

USER DEFINED CHARACTER SETS

- As many as you have RAM for (512 bytes each)
- Easy-to-use interface for defining your own character sets

~~$29.95~~

Till October 15, 1982

$24.95!

LUNAR RAIDERS

- Your mission is to fly over the surface and knock out the alien facilities, supplies, and ships without getting yourself shot down, or running into mountains
- Left, Right, Fire Controls
- 10 Speeds

ON A 272 X 192 DISPLAY!

~~$24.95~~

Till October 15, 1982

$19.95!

6502 SOFTWARE EMULATOR

- Learn 6502 machine language on your ZX-81
- User manual
- Plus introductory teaching guide to programming the 6502
- "Executes" 6502 machine code on your ZX-81
- Call Z-80 subroutines from your 6502 programs
- Limited 6502 Assembler (Supports all 6502 mnemonics but no symbols or macros)
- Graphic display of the 6502's status
- Single step, multiple break points, 7 different break point conditions
- "Fake" I/0 and interrupts

~~$29.95~~

Till October 15, 1982

$24.95!

6502 GRAPHIC HARDWARE EMULATOR

- All features of the 6502 Software Emulator
- Plus graphic display of the 6502 chip's hardware status
- Plus introductory teaching guide to the 6502 hardware

~~$39.95~~

Till October 15, 1982

$34.95!

User's manual for any of the above - $2.00 each
Teaching guide for software and hardware emulators - $5.00 each
REFUNDED ON PURCHASE OF PRODUCT

PRODUCT	QTY.	USER MANUAL	QTY.	TEACHING GUIDE	QTY.	TOTAL
CHARACTER CARTRIDGE/$24.95		$2.00				
LUNAR RAIDERS/$19.95		$2.00				
6502 SOFTWARE EMULATOR/$24.95		$2.00		$5.00		
6502 GRAPHIC HARDWARE EMULATOR/$34.95		$2.00		$5.00		
SHIPPING AND HANDLING/$4.50						$4.50

TOTAL ▶ ______

NAME ______

ADDRESS ______

CITY ______ STATE ______ ZIP ______

MAIL TO: SINCLAIR APPLICATIONS 5129 Alhambre Court
Norcross, GA 30093
(404) 921-4471

*Requires ZX-81.
Tested on U.S.
Version only.

Now available in the U.S. at the incredible price of only $119.95 (£62.95)

ACTUAL SIZE 3 X 2

THE 64K RAM PACK

- Fully built, tested and guaranteed.
- No additional power supply required.
- Black plastic case no larger than 16K RAM PACK.
- No wobble problems
- Fully compatible with printer etc.

The use of this memory is as follows:-

0 – 8K	Sinclair ROM
8 – 16K	RAM that is unaffected by NEW, LOAD, SAVE and can be used for storing machine code routines.
16 – 32K	BASIC Code, Display file, Variables, Machine Code
32 – 64K	Variables

ALSO AVAILABLE

16K RAM PACK at only $49.95 (£29.95)
including *FREE ALIEN ATTACK (7K -M/code)* — value $11.00 (£5.75)
As reviewed in 'Your Computer' — (March 1982) the leading U.K. Sinclair Publication. Almost 10.000 units sold in the U.K. so far.

PLEASE NOTE: All prices are fully inclusive and are applicable for export, including airmail postage. Payment may be made in sterling (Money Order available at your bank) or your own U.S. $ cheque, made payable to JRS SOFTWARE.
Despatch normally 7 days from receipt of order

JRS SOFTWARE

19 WAYSIDE AVENUE, WORTHING, SUSSEX. BN13 3JU Telephone: (0903) 65691

Explore the excellence of your **ZX81** with a

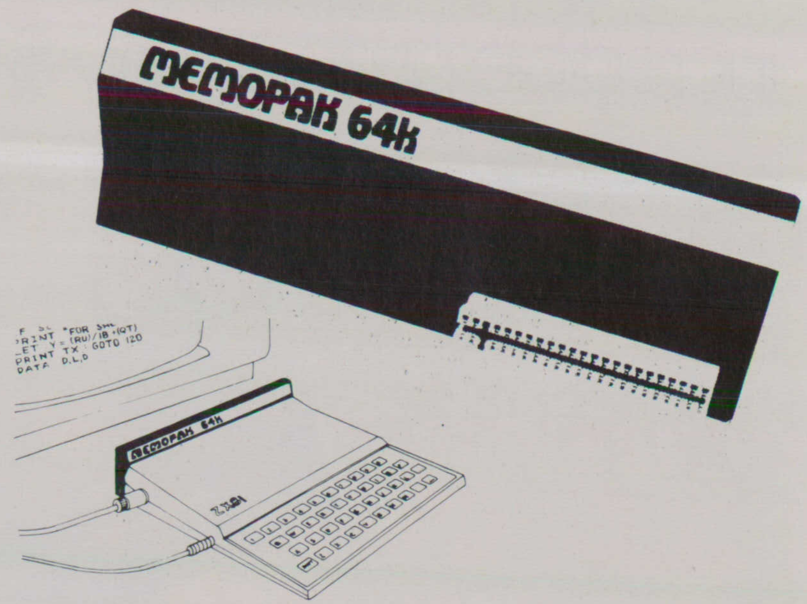
MEMOPAK

MEMOPAK

MEMOPAK

MEMOPAK

MEMOPAK



MEMOPAK 64k memory extension for $179.95

Give your diminishing memory more byte.

MEMOPAK 64K RAM $179.95

The Sinclair ZX81 has revolutionized home computing. The MEMOPAK 64K RAM extends the memory of ZX81 by a further 56K to a full 64K. It is neither switched nor paged and is Directly Addressable. The unit is user transparent and accepts such basic commands as 10 DIM A (9000). It plugs directly into the back of ZX81 and does not inhibit the use of the printer or other add-on units. There is no need for an additional power supply or leads.

Description of memory

0-8K . . . Sinclair ROM

8-16K . . . This section of memory switches in or out in 4K blocks to leave space for memory mapping, holds its contents during cassette loads, allows communication between programs, and can be used to run assembly language routines.

16-32K . . . This area can be used for basic programs and assembly language routines.

32-64K . . . 32K of RAM memory for basic variables and large arrays. With the MEMOPAK 64K extension the ZX81 is transformed into a powerful computer, suitable for business, leisure and educational use, at a fraction of the cost of comparable systems.

MEMOTECH

Memory Extention Specialists

Memotech Corporation
7550 West Yale Ave., Suite 220
Denver, Colorado 80227
Ph. (303) 986-0016

MEMOPAK 16K RAM $59.95

With the addition of MEMOPAK 16K, your ZX81 will have a full 16K of Directly Addressable RAM. It is neither switched nor paged and enables you to execute longer and more sophisticated programs and to hold an extended data base.

The 16K and 64K Memopaks come in attractive, custom-designed and engineered cases which fit snugly on to the back of the ZX81 giving a firm connection.

Free service on your MEMOPAK

Within the first six months, should anything go wrong with your MEMOPAK, return it to us and we will repair or replace it free of charge.

Try MEMOPAK with no obligation

You can use our MEMOPAK in your home without obligation. After 10 days if you are not completely satisfied, simply return it for a full refund.

Coming soon . . . . .

A complete range of ZX81 plug-in peripherals:

- MEMOTECH Hi-Res Graphics
- MEMOTECH Digitising Tablets
- RS232 Interface
- Centronic Interface and Software Drivers

All these products are designed to fit "piggy-back" fashion on to each other and use the ZX81 power supply.

Further information forthcoming.

Memotech Corp. 7550 W. Yale Ave. Suite 220 Denver, Colo. 80227

Yes! I would like to try the Memopak. I understand that if I'm not complete satisfied, I can return it in 10 days for a full refund.

	Price †	Qty.	Amount
Memopak 64k RAM	$ 179.95		
Memopak 16k RAM	$ 59.95		
Shipping and Handling	$ 4.95	———	$ 4.95
		Total	

☐ Check ☐ Visa ☐ MC

Act. No. ____________ **Exp.** ______

† U.S. Dollars

Name ______________________

Street ______________________

City ______________ **State** ______ **Zip** ______

SYCO 7-8

The magazine for Sinclair users

SYNC®

July/August 1982 Volume 2, Number 4

Volume 2, Number 4

SYNC (USPS: 585-490: ISSN: 0279-5701) is published bi-monthly by Ahl Computing, Inc., a subsidiary of Ziff-Davis Publishing Company. David Ahl, President; Elizabeth B. Staples, Vice President; Selwyn Taubman, Treasurer; Bertram A. Abrams, Secretary. 39 East Hanover Avenue, Morris Plains, NJ 07950. Second class postage paid at New York, NY 10001 and at additional mailing offices.

Subscription rates: USA: One year (6 issues), $16; Two years (12 issues), $30; Three years (18 issues), $42. Canada: $3 per year additional. Other foreign: $5 per year additional.

For *SYNC* advertising information, contact Jim Beloff, *SYNC* Advertising Sales Manager, Ziff-Davis Publishing Company, One Park Ave., New York, NY 10016 (phone 212/725-4216).

All other correspondence should be addressed to: *SYNC*, 39 E. Hanover Ave., Morris Plains, NJ 07950. In U.K., *SYNC*, 27 Andrew Close, Stoke Golding, Nuneaton CV136EL.

Postmaster: Send address changes to *SYNC*, PO Box 789-M, Morristown, NJ 07960.

MEMBER MITA

Photos from *Star Trek II: The Wrath of Khan* courtesy of **Bruce Birmelin/Paramount Pictures.**

letters

LSCROLL

Dear Editor:

For entering machine code routines like *LSCROLL* (*SYNC* 2:2) it is helpful to use the 8K ROM's string concatenation abilities. The procedure involves breaking up the one long string in line 10 into a series of shorter strings which are entered line by line.

Set the lines up as follows:

10 LET D$="(begin entering mc numbers)"

20 LET D$=D$+"(continue entry)"

30 LET D$=D$+"(continue entry)"

etc.

Thus each new line includes all the preceding lines, and we end up with the whole string from the short strings. This makes it much easier to correct the entries since each line can be individually EDITed.

Also, the following lines give several advantages in SAVEing:

```
1 GOTO 3
2 SAVE "name of program"
```

You now do not have to type in SAVE every time when you want to make several SAVEs. Simply type in GOTO 2. The program will SAVE itself, then RUN itself. If you want to make more than one SAVE, you must change the last character of the name of the program to normal video from inverse (for some reason, the system does that).

The real advantage comes when LOADing because the program will now start itself!

I have two other hints. DATA statements could be simulated using strings; the 8K's VAL function could be used to remove VALues. It usually makes entering MC easier to show a zero as 0 with a / through it.

Erik Sawyer
1213 Patriot Dr.
Slidell, LA 70458

Dear Editor:

Douglass Sharp's *LSCROLL* program (*SYNC* 2:2) is a very useful expansion of the ZX81's graphics capabilities. As written it requires the 16K RAM, but a few modifications allow *LSCROLL* to perform its magic with the 2K RAM since it occupies only 172 bytes. (The numbers correspond to the steps in the original article.)

1) Rather than have D$ take up valuable space within the *LSCROLL* program, enter D$ first without a line number as follows:

```
LET D$= "3AFE47FE16DA61473E0"
LET D$=D$+"032FE473AFF47D600CA7147FE15D2714"
LET D$=D$+"7C376473E1632FF47F53AFF4747F13AF"
LET D$=D$+"E4780FE16DA954726002E163AFE474F0"
LET D$=D$+"600ED427D32FF47ED5B0C400EFFF53AF"
LET D$=D$+"E476FF1260006001AFE76CAAF4713C3A"
LET D$=D$+"5470CA7E5ED42E1CABC4713C3A547060"
LET D$=D$+"0F53AFF474FF1C5D5E1237EFE76C2C74"
LET D$=D$+"7E5A7ED522B7D4DD600E1C2E047E5D1C"
LET D$=D$+"3F5473DC2E84713C3F14706000B13D5E"
LET D$=D$+"123EDB03E001213C10B79D600C2C447C"
LET D$=D$+"90000"
```

These lines have the same length as those in Figure 1 in *LSCROLL*, but the entries have been changed to conform to the addresses appropriate to the 2K RAM. Furthermore, the arrangement makes checking the entries easier.

Enter lines 4 and 20-80 as in Figure 1 but change the address in line 50 to 18259. Then add:

```
10 CLS
90 STOP
100 SAVE "LSCROLL"
110 PRINT "GOTO 4 (TO RUN)"
```

2) SAVE by entering GOTO 100 (NEVER use RUN with this program for it will erase D$).

3) Enter NEW.

4) Enter POKE 16388,84 and POKE 16389,71. Enter NEW.

5) LOAD "LSCROLL" and then enter GOTO 4 (as the screen reminder indicates). In a few seconds the number 21093 will appear. If it does not, check for an error in D$.

To execute, use RAND USR 18260. To change the values of XX, YY, and CC (Figure 3), POKE the appropriate values into 18430, 18431, and 18418 respectively.

Figure 5 should be corrected as follows:

7F66 with 32614, not 32615
7F95, not 7FA5, with 32661
7FB4, not 7FC4, with 32692
7FE5 with 32741, not 32740

Harold Miller
Mountainview
Route 3
Clarkesville, GA 30523

Software Breakthrough!

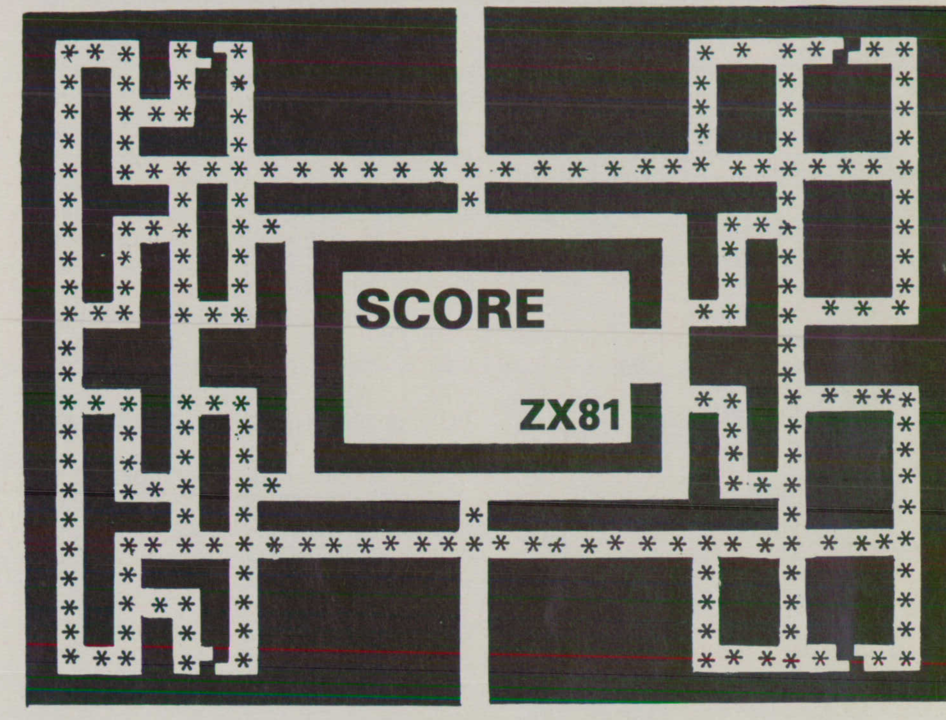
✱ ZX-Man $14.95

✱ Joystick $14.95

✱ 16K Sinclair Memory $69.95

✱ Keyboard $74.95

full-size assembled w/leatherette case



RUNS WITHOUT ANY HARDWARE MODIFICATIONS!

1) Use with or without our joysticks.
2) Real time Animation
3) Loads of Fun
4) Auto Run
5) 100% Machine Code
6) No Screen Flicker
7) ZX80 and ZX81 compatable
8) 16K Required

THE INNOVATORS ARE COMING

Call or Write for more information

American Express, Visa, Master Charge, Money order, pc (allow 2-3 weeks to clear)
Please add $2.50 postage & handling
Illinois residents please add 5% Ill. sales tax

Visit our Showroom 10:00 - 6:00 Monday thru Saturday

AMERICAN EXPRESS Cards Welcome

VISA

MICROCOMPUTERS PLUS, INC.
349 EAST MAIN ST.
GALESBURG, IL 61401
PHONE: (309) 342-9572

master charge THE INTERBANK CARD

Flag Use

Dear Editor:

I just saw a copy of *SYNC* 2:1 with Philip Gervais's tip on flag use:

100 LET A=ABS(A-1)

A simpler and shorter statement to do the same thing is

100 LET A=1-A

Keep up the good work; I enjoy the magazine.

David Schultz
Total Information Services, Inc.
PO Box 921
Los Alamos, NM 87544

Dear Editor:

An easier way to change a flag is

LET A=NOT A

This is also 10 bytes shorter.

Tim Goldingham
11 Furze Platt Rd.
Maidenhead SL6 7ND
United Kingdom

Key and Token Expressions

Dear Editor:

As Richard W. McDaniel points out (*SYNC* 1:6), the use of key and token expressions in PRINT and REM statements can save typing time and memory. There is, however, another way to insert these into a program which is sometimes easier.

Entering THEN gets the ZX80 and ZX81 into condition to accept keystrokes as tokens. Suppose that you want to define A$ as "STOP SEEING YOUR NAME IN PRINT FOR THE NEW YEAR". You would key it in as follows:

10 LET A$="THEN STOP (delete THEN) SEEING YOUR NAME IN THEN PRINT (delete THEN) THEN FOR (delete THEN) THE THEN NEW (delete THEN) YEAR"

This is easier than it looks. The "DELETE THEN" procedure uses only three keystrokes: backspace (shifted 5), delete (shifted 0) and forward space (shifted 8).

Basil Wentworth
1413 Elliston Dr.
Bloomington, IN 47401

Hams and the ZX81

Dear Editor:

I am a delighted ZX81 owner. I am also an amateur radio operator (ham).

Rather than trying to communicate with other ZX81 owners by newsletters, I would be interested in comparing notes more directly: I propose to organize a ZX81 network on radio. This would entail deciding on the frequency, day of the week, and time of day. After this we would conduct our regular communications on a conversational basis.

Other combination radio amateur operators and ZX81 users are asked to contact me so that we can get organized soon. And, if you have a "ham" on the *SYNC* staff, it would be wonderful if he would meet with us on the air to field some of the questions which are sure to arise!

Dean Sturm, K8CYW
1823 Enslow Blvd.
Huntington, WV 25701

Ed.—We do not have a "ham" on our staff, but we have heard from some operators interested in combining these two interests. One ZX81 Ham network is already organized and publishes a newsletter QZX (contact Martin Irons, K2MI, 46 Magic Circle Dr., Goshen, NY 10924, for details). We would appreciate hearing of any others.

ZX80 ROMs

Dear Editor:

In Michael Rubesch's "4K/8K ROMs in One ZX80" (*SYNC* 2:2) the reference to "pin 28" of the modules rather than to "pin 28" of the IC socket was a bit confusing. Later PC boards do have a 28-pin socket for the ROM while the ROMs have only 24 pins. Also I have found that the earlier ZX80 boards have only 24-pin sockets if they have sockets at all. The power pin then is pin 24 of the ROM module which must be inserted at pin 28 of the later sockets.

Robert D. Hartung
PO Box 125
Palmyra, NY 14522

Hardware Suggestion

Dear Editor:

In this avalanche of new hardware for the ZX81 I wonder how many manufacturers realize that many of us bought the ZX81 because it is small. I can put it and my equally small cassette recorder into a thin slipcase and use it anywhere a TV is available. No other low or medium priced machine offers that degree of portability. Every manufacturer of hardware for the ZX81 should keep this in mind and should include the size of the product in ads. I favor stuffing as much memory and other extensions inside the case as possible.

Harold Miller
Mountainview
Rt. 3
Clarkesville, GA 30523

Inventory

Dear Editor:

The *Inventory* program (*SYNC* 1:6) was both enjoyable and useful. Here is an improvement to SAVE the program (and variables) as directed by the prompt in line 2018:

Change line 2020 to:

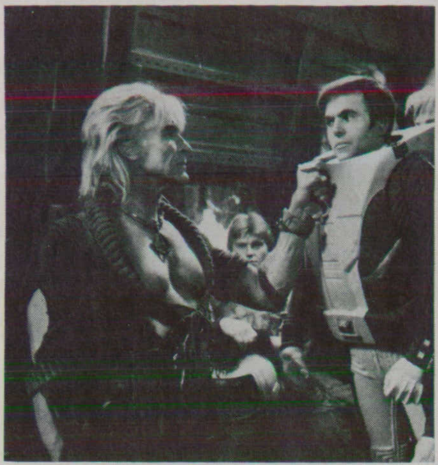
```
2020 IF S$="C" THEN SAVE "INVENTORY"
```

Add:

```
2019 INPUT S$
2021 STOP
```

Thanks for a great magazine!

TSGT Chuck Taylor
Diyarbakir
Turkey



Bruce Birmelin/Paramount Pictures.

KAYDE

KAYDE Electronic Systems
ZX80/1
ZX KEYBOARD WITH REPEAT KEY

$75.90

Fully cased keyboard $75.90
Uncased keyboard............. $55.90
Keyboard Case.................. $21.90

This is a highly professional keyboard using executive buttons as found on top quality computers. It has a repeat key and comes complete in its own luxury case. This is a genuine professional keyboard and should not be confused with toy keyboards currently available on the market

KAYDE 16K RAM PACKS

$59.90

The 16K RAMPACK simply plugs straight into the user port at the rear of your computer. It is fully compatible with all accessories and needs no extra power and therefore it will run quite happily on your Sinclair power supply. It does not over-heat and will not lose memory at all. As you may know, some makes go down to 11K after being on for a while.

This 16K RAMPACK is very stable and will not wobble or cause you to lose your programme. It comes fully built and tested with a complete money-back Guarantee.

KAYDE FLEXIBLE RIBBON CONNECTOR

Stops movement of RAM PACK and other accessories
(Not needed with a *KAYDE* RAMPACK)

$25.90

KAYDE 4K GRAPHICS BOARD

$59.90

The *KAYDE* Graphics Board is probably our best accessory yet. It fits neatly inside your ZX81. It comes complete with a pre-programmed 2K Graphics ROM. This will give nearly 450 extra graphics and with the inverse makes a total of over nine hundred.

The *KAYDE* Graphics Board has facilities for either 2K of RAM (for user definable graphics), 4K of ROM or our 4K Tool Kit Chips that will be available shortly. All the graphics are completely software controlled, therefore they can be written into your programmes. Here are a few examples: **A full set of space invaders — Puckman — Bullets, Bombs — Tanks — Laser Bases and Alien Ships.**

NO EXTRA POWER NEEDED

KAYDE 16K GRAPHICS BOARD SOFTWARE

$11.90

PECKMAN: The only true ZX version of the popular arcade game.
Centipede: "In all I think this is the best presented moving graphics program I've yet seen." Phil Garratt, Interface.
SPACE INVADERS: The best version available anywhere. Graphics software can only be used with a graphics board

KAYDE 16K 81 SOFTWARE

$11.90

Centipede: "In all I think this is the best presented moving graphics program I've yet seen." Phil Garratt, Interface.
3D/3D Labyrinth: A Cubit Maze that has corridors which may go left, right, up, down. Peckman (the latest addition in 81 games).

WHY WAIT TO PAY MORE —
FAST, IMMEDIATE DELIVERY

Post to: Dept SY1
Kayde Electronic Systems Ltd
The Conge
Great Yarmouth
Norfolk NR30 1PJ ENGLAND
Tel: 0493 57867 (Dept. SY1)

Please pay by
International money order or certified cheque for immediate delivery.

Please send me .... ZX Keyboards with repeat key $75.90 each
Please send me .... 16K RAMPACKS $59.90 each
Please send me .... Flexible ribbon connectors $25.90 each
Please send me .... 4K graphics boards $59.90 each
Please send me .... 16K graphics board software $11.90 each
Please send me .... 16K 81 software $11.90 each

I enclose ____________________

Name ____________________

Address ____________________

____

____

Please add $5 p&p for all hardware and $2 for all software. Please make cheques payable to Kayde Electronic Systems Ltd.

sync notes

Paul Grosjean

SYNC in Outer Space

Space games lead the list of the games that *SYNC* receives. This we would expect since shoot-the-aliens games have become standard fare in the computer game field.

Certainly the computer is ideal for playing such games. As we have seen in recent years, modern instruments of war depend heavily upon computer controlled technology. Hitting the target used to be the result of an almost intuitive interpretation of a host of variables which was then put into action by eye/muscle coordination. The most important factor in hitting the target seemed to be luck—another name for the unknown or inmeasureable variables. The lack of instruments to measure and calculate led to the use of massive fire power in the hope that something would hit the target eventually.

Now, however, hitting the target is a matter of computation based on precise measurement with lasers or radar. Since space warfare involves speeds, distances, power, and logistics beyond our experience, it is necessarily computer warfare.

While the technology was developing that made space warfare possible, our imaginations were also being prepared. Science fiction writers such as E. E. Smith, A. E. van Vogt, and Isaac Asimov prepared us to think of conflict between humans and aliens and between humans and humans on a galactic scale.

Movie features and serials provided our imaginations with the visual and sound effects. The famed radio broadcast of H. G. Wells's *War of the Worlds* in 1938 raised conflict with the aliens high in the national consciousness. Finally TV put space into every living room. The old movies were shown again, the new movies reached larger audiences, and programs were developed around science fiction themes, most notably *Star Trek*. A space vocabulary has developed and become part of our daily vocabulary.

We have the building blocks for story telling, and the computer has become the means of telling the story, but now the player participates in unfolding the story. LET A=(something) is necessary to the computer, but to the player LET A may really equal a space ship, an alien, a minefield, a forest, a castle, a maze—whatever the imagination desires. A given computer program can provide a host of stories without changing a line in it. We only need to change the terms we assign to the variables and the relationships we have instructed the computer to work out.

In order to simulate some of the unknowns of real life, we even introduce random numbers and elements into the program. *SYNC* receives from time to time games that are purely the "battle of the random numbers." In general, unless the program illustrates some significant programming techniques, these are rejected because the player's skill and judgment are vital parts of any satisfying game. If the player wins, he can claim the victory for his skill or ability; of course, if he loses, he can still blame bad luck.

In this issue we have gathered some of these games into a theme section "SYNC in Outer Space." We are grateful to Paramount Pictures for providing photos from *Star Trek II* to highlight our theme.

WANT TO FIX YOUR SINCLAIR? ADD MORE MEMORY? ADD A BUSS? WE GOT YOUR PARTS.

We carry all the parts you need to repair your Sinclair with the exception of the Sinclair Logic Unit, the ROM, and the Printed Circuit Board. In addition we carry OK. WELLER. and XCELITE tools, HAMMOND, and PAC-TEC cases and VECTOR breadboarding and prototyping supplies. We stock resistors, capacitors, integrated circuits, descrete solid state devices, switches, sockets, connectors etc. etc. Below is a very small sampling of our products. If you see what you need give us a call, if you don't write for our free catalog.

INTEGRATED CIRCUITS

VOLTAGE REGULATORS

POSITIVE		NEGATIVE	
7805	+5V	7905	-5V
7812	+12	7912	-12
7815	+15	7915	-15
7824	+24	1.25 ea.	

HEAT SINK 25¢

LM323K 5VOLT 3AMP 3.95
HEAT SINK 75¢

C.P.U. & SUPPORT

Z80A CPU 9.45
Z80A PIO 7.65
Z80A CTC 7.65
Z80A DMA 19.95
Z80A SIO/2 ..... 23.95

MEMORY

2114	1KX4 STATIC	4.00	8/28.00
4118	1KX8 STATIC	8.25	8/58.00
6116	2KX8 STATIC	14.95	8/125.00
4116	16KX1 DYNAM	3.50	8/24.00
2716	2KX8 EPROM	12.00	8/84.00
2732	4KX8 EPROM	19.95	8/135.00

TTL

PARTIAL LISTING

7400..... .24	74LS00..... .25
7403..... .24	74LS03..... .25
7409..... .28	74LS09..... .30
7490..... .35	74LS90..... .45
74148.... .75	74LS138.... .70
74154.. 1.25	74LS154.. 1.75
74190.... .75	74LS190.... .85
74LS244 1.00	74LS245.. 1.75

CMOS

PARTIAL LISTING

74C00..... .40	4001.... .30
74C02..... .40	4011.... .30
74C30..... .40	4016.... .60
74C90.... 1.35	4020.... 1.20
74C154.. 4.25	4071.... .30
74C240.. 2.35	4511.... 1.15
74C244.. 2.35	4553.... 3.95

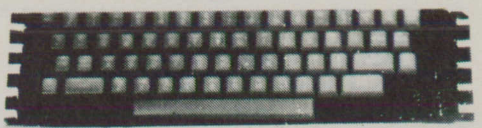
LINEAR

MC3401 QUAD OP-AMP .... .50
MC1555 TIMER.................. .45

	BI-FET OP-AMPS	LOW NOISE
SINGLE	TL081... .75	TL071... .90
DUAL	TL082. 1.25	TL072.. 1.50
QUAD	TL084.. 2.25	TL074. 2.50

FULL—SIZE KEYBOARD

INCREASE YOUR COMPUTING TIME BY DECREASING TYPING TIME ADD A FULL-SIZE KEYBOARD TO YOUR ZX-80/81 OR MICROACE. YOUR FINGERS WILL LOVE YOU FOR IT.



OUR KEYBOARD HAS 62 KEYS SO YOU CAN ADD LOTS OF EXTRA FUNCTIONS. WE WILL EVEN SEND YOU PLANS ON HOW TO ADD SUCH USEFUL THINGS AS AUTOMATIC SHIFT ON CERTAIN FUNCTIONS, AUTOMATIC REPEAT AND RESET. EACH KEY HAS GOLD CONTACTS AND IS FIRMLY MOUNTED TO A BLACK METAL PLATE WHICH MEASURES 15-1/2" BY 4" AND DOUBLES AS AN ATTRACTIVE FACE PLATE.

BEST OF ALL YOU CAN GET YOURS IN KIT FORM, COMPLETE WITH INSTRUCTIONS, OR WIRED AND READY TO GO. EITHER WAY WE WILL INCLUDE THE EXSPANSION PLANS.

KEYBOARD KIT. 34.95 WIRED & TESTED..49.95

SINCLAIR CONNECTOR

GOLD CONTACTS WIRE WRAP OR SOLDER TAIL $7.50

R. F. MODULATOR

USA VHF 5.95

DATA CASSETTES

\$1.75

10 for \$15.00

Made for Datapoint by Maxell
90 min. with case
5 screw high strength shell

JOYSTICK

4 - 10K Linear pots
1 - V2 metal handle
2 - V8 square

$5.95

YOUR ULTIMATE POWER SUPPLY

WHY WASTE TIME BUILDING NEW POWER SUPPLIES WHEN YOU WANT TO EXPAND? BUY AN ULTIMATE SUPPLY AND SPEND ALL YOUR TIME ON PROJECTS NOT P.S.

MULTIVOLTAGE
FOR DYNAMIC RAMS ETC.
INPUT ___ 120 - 220 VAC
OUTPUTS +5VDC--5AMP
+12VDC__1AMP
−12VDC−1AMP
12VAC−1/2 A.

5-3/4X7-1/2X4 H MOD CPS-59.................50.00
PARTS & PLANS TO CONVERT 12VAC TO +24 FOR EPROM PROGRAMING.....ADD...5.00

SINGLE VOLTAGE
FOR CMOS & TTL LOVERS
INPUT-------- 120 - 220 VAC
OUTPUT----- +5VDC---9AMP
DIM.------9-1/2 X 5 X 5
MOD CPS−169.............50.00

ALSO AVAILABLE
120 IN 5VDC--3AMP MOD SPS 30-5...24.95
120 IN 5.8VDC--1.2AMP MOD SPS 1-5......5.95

SOLDER

$7.95

KESTER 60/40
Rosin core .020 dia.
1 Pound Roll

IC SOCKETS

#PIN	SOL	WW
8	.20	.65
14	.25	.70
16	.30	.75
18	.36	.95
20	.40	1.10
22	.45	1.20
24	.50	1.25
28	.55	1.70
40	.75	1.95

SOLDERING IRONS

WELLER
WP25 120V 25W 17.03
WP40 120V 40W 20.93
PENCIL THIN IRON (NOT SHOWN)
WM120 120V 12W 19.47

SOLDERING TOOL STAND

STAND WITH SPONGE, BASE & RECEPTACLE.

FITS WP-25 WP-40 W-60
MOD PH-60...........10.14

FITS WM-120
MOD PHM-120.......10.14

WIRE WRAP WIRE

KYNAR #30 WIRE WRAP WIRE

PRECUT TO LENGTH THEN 1" STRIPPED FROM EACH END.

WE GOT IT IN RED, YELLOW, BLUE,BLACK,WHITE & GREEN

LENGTH	100/BAG	1000/BAG
3.0"	1.45	7.50
3.5"	1.55	8.25
4.0"	1.65	9.00
4.5"	1.75	9.75
5.0"	1.85	10.25
ADD'L 0.5" UP TO A TOTAL OF 10.0"	.10	.75

#30 BY THE SPOOL
100' ...3.95 500' ..11.95
1000' ...19.95

HOBBY WRAP TOOL

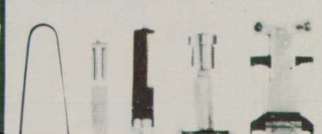
AUTO-INDEX
ANTI-OVERWRAP
MODIFIED WRAP
TOOL WITH BIT
BW2630B. 27.95

DESOLDERING PUMP

EASY ONE HAND OPERATION
ALL METAL CONSTRUCTION
REPLACEABLE TEFLON TIP
REGULATED SUCTION

DSP1 ....11.85

INSERTION / EXTRACTION TOOLS



INS1416 INSERT TOOL 4.35

MOS SAFE INSERT TOOLS
MOS1416 14-16 PIN.....9.95
MOS2428 24-28 PIN.....9.95
MOS40 36-40 PIN....10.95

EXTRACTION TOOLS
EX1 14-16 PIN.....1.49
EX2 24-40 PIN. ...9.95

WK7 COMPLETE KIT ...37.95

HAND WRAP TOOL

WRAPS AND UNWRAPS

BUILT-IN#30 WIRE STRIPPER

WSU30....REGULAR..........7.25
WSU30M MODIFIED.........8.25

VACUUM VISE

VACUUM BASED LIGHT DUTY VISE
1-1/2" WIDE JAWS
1-1/4" TRAVEL

VV1.................4.15

CLIP AND STRIP

CUT AND STRIP#30 WIRE IN ONE OPERATION.TAKES A LOT OF THE WORK OUT OF WIRE WRAPPING.
CAS130................2.25

TERMS

Check, money order, VISA, or Mastercharge. Add $3.00 for shipping and handling. Texas residents add 5% state sales tax. Foreign orders add $10.00 for shipping; U.S. funds only. Limited quanities on some items.

ALTEX ELECTRONICS

618 W. Sunset
San Antonio, Texas
78216

1—512—828—0503

try this

"Try This" features short programs to show off your computer, impress your family and friends, and tickle your imagination when *SYNC* arrives at your place. Send your contributions to: Try This, SYNC, 39 E. Hanover Ave., Morris Plains, NJ 07950.

4K ROM

Type in the following program. Then RUN and NEWLINE. Type in any character or symbol when the quotation marks appear and hit NEWLINE again. Observe the results. Repeat until you get back to program. Note the program.

Our thanks to:

Ken Berggren
104 Ridgeway Ave.
Louisville, KY 40207.

```
 5 PRINT 1
10 POKE 16427,1
15 DIM A(9)
20 LET B=128
25 INPUT C$
30 GOTO 5
```

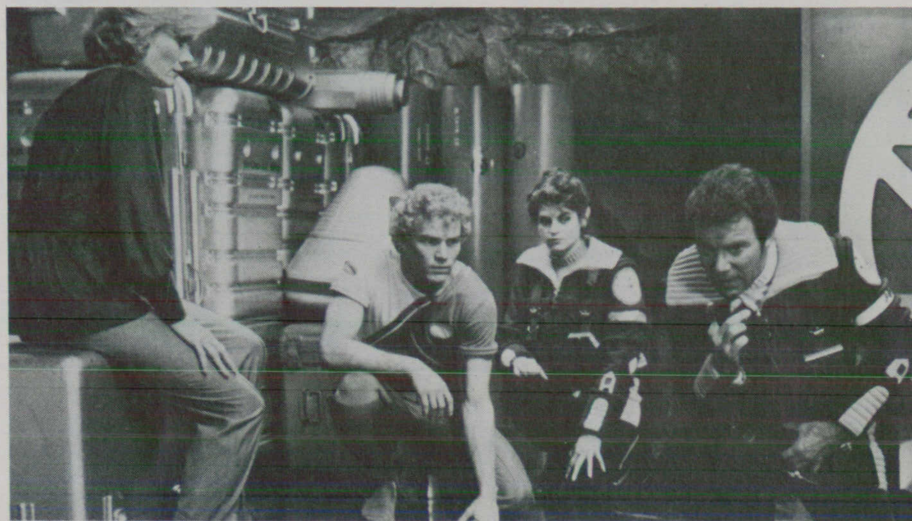
8K ROM

Type in the following program. Then hit RUN and ENTER. Observe the results. If you like, you can substitute other characters in the print statements in Lines 20 and 50.

Our thanks to:

Ross A. Rainwater
305 Regal Drive
Lawrenceville, GA 30245

```
10 FOR X=10 TO 15
15 SCROLL
20 PRINT TAB (X);"***"
30 NEXT X
40 FOR X=15 TO 10 STEP -1
45 SCROLL
50 PRINT TAB (X);"***"
60 NEXT X
70 GOTO 10
 5 PRINT 1
10 POKE (16427),1
15 DIM A(9)
20 LET B=128
25 INPUT C$
30 GOTO 5
```



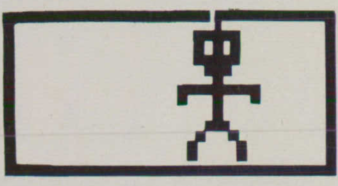
Bruce Birmelin/Paramount Pictures.

SOFTSYNC, INC.

TS 1000 & ZX81 SOFTWARE

NEW RELEASES!

H-A-N-G M-A-N



Remember when you played Hangman as a child? We've created a learning tool for you and your children which contains 7 versions of Hangman on the ZX81.

Your child can play against the computer's 5 pre-recorded libraries of categorized words, (each word up to 17 letters) Animals, Science, Countries of the World, States of America, and Fruits and Vegetables.

As a special feature during the game, each program will pick a word from its own library, then compare it with the last 50 words played so you will not get repeat words for at least an hour of play. Also the computer keeps a running account of how many words you get right and wrong.

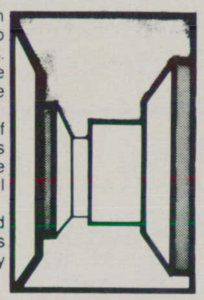
In the 6th version, you or your child may also play against an opponent. The computer requests players names, keeps score and generates the Hangman graphics.

The seventh option allows you to create and save your own versions with up to 500 words each. This is ideal for parents to create special spelling drills for their children.

Hangman is a great spelling and vocabulary tool for youngsters 6-12 years old, although its the kind of game adults have been known to indulge in too.

16K $14.95

SUPERMAZE



SUPERMAZE is a three-dimensional maze program with extraordinary full screen graphics that challenges you to work your way through a complex labyrinth of corridors. Enter SUPERMAZE at your own risk and carefully choose the right combinations of passageways to reach the end in the fewest possible moves.

This graphically illustrated adventure game is full of hazards to impair your ability to get through it. For example as you wander through the endless corridors, you have to be careful because there are hidden trap doors you can fall through, finding yourself in another part of the maze.

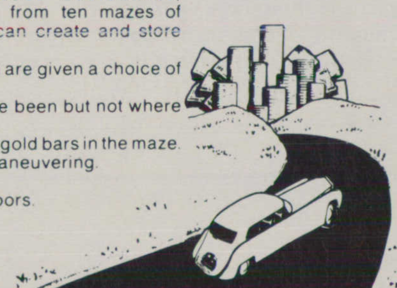
But SUPERMAZE also features some helpful devices to aid your journey. You can have a compass to guide you or stones to drop along the way to let you know if you have already passed through a section of the maze.

Written with many machine code subroutines, SUPERMAZE allows you to choose from ten mazes of increasing difficulty. In addition you can create and store mazes of your own design.

At the beginning of the program, you are given a choice of options to play including:

- A map option to see where you've been but not where you are going.
- An option to find and take with you gold bars in the maze. This helps you to develop skill in maneuvering.
- Stones to drop along the way.
- Optional randomly placed trap doors.
- The compass to guide you.

16K $14.95



ROAD TO RICHES

People have always said, "If only I had some money to play with, I could turn it into a fortune...I could be rich!" Well this is your dream come true -- an opportunity to show your prowess at making wise investments because you have just been given a million dollars -- to play with!

Can you turn it into ten million? Or will you lose it all on a risky investment? It's up to you.

ROAD TO RICHES is an entertaining and educational computer game in which up to four players are given opportunities to invest in business ventures in an attempt to increase their assets.

The program simulates a fast-moving world of investment in which you are confronted with all kinds of investments ranging from conservative real estate opportunities to highly risky Broadway shows.

The program carefully weighs the risk of each investment and your chances for winning or losing. Like real life, low risk usually means (but not always) low yield while high risk can mean hitting the jackpot or financial disaster.

Some investments give you more working capital while others give you equity with annual returns.

Pit yourself against a wheeler-dealer and see if, over a period of seven years, you can come out on top. ROAD TO RICHES is fun and informative, realistically portraying the risk and skill of high finance.

16K $14.95

PLUS MANY MORE ZX81 PROGRAMS!

WRITE FOR FREE CATALOG

Please send me:

___copies of **HANGMAN @ $14.95 ea**
___copies of **SUPERMAZE @ $14.95 ea**
___copies of **ROAD TO RICHES @ $14.95 ea**

Please indicate method of payment —
☐ Check, money order (no cash please)

Charge to:
☐ VISA/Bankamericard
☐ Mastercharge

No ________________ Exp. date ________

Signature ____________________

Add $1.50 shipping and handling.
New York residents add sales tax.

SEND CHECK OR MONEY ORDER TO:

SOFTSYNC, INC.

P.O. Box 480, Murray Hill Station, New York, N.Y. 10156 (212) 685-2080

NAME ____________________

ADDRESS ____________________

CITY ____________________

STATE/ZIP ____________________

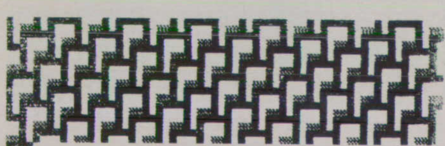
just for fun

Generally *SYNC* prefers articles in some depth to help you get more out of your computer. However, we receive many short programs that illustrate a point, demonstrate a technique, or show something the reader has found interesting. "Just for Fun" shares these programs with you. If you learn something, great. If you have some fun, great. If you have some that you want to share, send them to: Just for Fun, SYNC, 39 E. Hanover Ave., Morris Plains, NJ 07950.

Graphics Loop I

Bernard Bush

One technique for squeezing more into the limited memory of the basic ZX81 is to use a loop. The following program illustrates the method applied to create a continually changing graphics display. Enter the program. Then be sure you are in SLOW mode. Hit RUN and ENTER and enjoy the show.



```
  10 REM GRAPHICS LOOP
  20 LET A=INT (RND*10)+1
  30 LET B=INT (RND*10)+1
  40 LET C=INT (RND*10)+1
  50 LET D=INT (RND*10)+1
  60 LET E=INT (RND*10)+1
  70 PRINT
  80 PRINT
  90 PRINT
 100 PRINT
 110 FOR J=1 TO 58
 120 PRINT CHR$ A;
 130 PRINT CHR$ B;
 140 PRINT CHR$ C;
 150 PRINT CHR$ D;
 160 PRINT CHR$ E;
 170 NEXT J
 180 UNPLOT J,J-16
 190 GOTO 10
 200 REM STOP WITH BREAK
```

Bernard Bush, Rt. 2, Mansfield, MO 65704.

Graphics Loop II

James Grosjean

The same screen display developed in "Graphics Loop I" can be achieved by the following program which establishes the loop through a technique suggested by Ken Berggren in his "Handling Strings from another DIMension" in this issue.

```
  10 FOR I=1 TO 5
  20 LET A=RND*9+1
  30 POKE 16537,38+I
  40 NEXT I
  50 POKE 16537,38
  60 PRINT AT 5,0
  70 FOR J=1 TO 58
  80 PRINT CHR$ A;CHR$ B;CHR$ C;
CHR$ D;CHR$ E;
  90 NEXT J
 100 RUN
```

James Grosjean, 50 Kings Rd., Chatham, NJ 07928.

Draw It

Robert S. Boynton

Many drawing programs require you to encode a picture and then display it, but this short program allows you to draw and edit a detailed picture even in 1K. Enter the program, be sure you are in SLOW mode, and press RUN and NEWLINE. A black pixel will be displayed at center screen. You can move the pixel by using the arrow keys (no need to shift). If you make a mistake or you want to move without leaving a mark, press RUBOUT and the pixel will flash. You can now move it to a new location or backwards to erase. Press RUBOUT again to return to plotting. You can also put any keyboard character into your picture by pressing the EDIT key (no shift), and insert the character you want in the quotation marks. Press NEWLINE and the character will appear at the pixel location. The RUBOUT feature works on the printed characters. Do not go beyond the normal machine plotting area or error B will result. If you do not want to use the characters, you can omit lines 110 and 130 to 150.

```
  10 LET Z=0
  20 LET X=32
  30 LET Y=22
  40 PLOT X,Y
  50 IF Z THEN UNPLOT X,Y
  60 IF INKEY$="5" THEN LET X=X-
1
  70 IF INKEY$="8" THEN LET X=X+
1
  80 IF INKEY$="6" THEN LET Y=Y-
1
  90 IF INKEY$="7" THEN LET Y=Y+
1
 100 IF INKEY$="0" THEN LET Z=NO
T Z
 110 IF INKEY$="1" THEN GOSUB 13
0
 120 GOTO 40
 130 INPUT A$
 140 PRINT A$
 150 RETURN
```

Robert S. Boynton, 189 David Humphreys Rd., Derby, CT 06418.

BYTE-BACK modules

64-K MEMORY $119.95 KIT

WIRED and TESTED $129.95

IN STOCK!

SAME DAY SHIPMENT!

BYTE-BACK'S M-64 extends the memory of your ZX81 or Timex-Sinclair 1000 to a full 64-K. It's user transparent. It plugs directly into the back of the ZX81 and has an expansion port to allow you to still use a printer. No extra power supply is required. It has all standard features **plus** the area from 8-16-K can be switched out in **2-K** increments for memory mapped peripherals, PROMS, etc. Same proven reliability as our M-16 with thousands in use.

INSTANT INFORMATION WITH BYTE-BACK'S MD-1

MODEM *only* $99.95 KIT

WIRED and TESTED $119.95

Use your phone to connect your "LITTLE" ZX81 to the "LARGEST" computer networks in the world. With **BYTE-BACK's MD-1 MODEM** connected all you do is dial a phone number (usually local), press a few keys and watch the data appear on your TV screen. (Software is included) This **MODEM** can be used in either the "originate" or "answer" mode with selectable baud rate.

You can have immediate access to:

UNIVERSITY COMPUTERS, DOW JONES, UPI, AND MORE !

As an extra bonus an RS-232 port is provided to allow you to drive all standard RS-232 peripherals. (75 to 9600 Baud)

EXPAND YOUR 16K SYSTEM

(M-16) $59.95 KIT

WIRED and TESTED $69.95

If you have a Sinclair 16K RAM module and need more memory, expand it to 32K and beyond by using **BYTE-BACK M-16 MEMORY MODULES.**

You can't connect two Sinclair 16K RAM modules together, but you can connect one Sinclair 16K and one or more **BYTE-BACK 16K** modules to get all the memory you need.

THOUSANDS IN USE WITH PROVEN RELIABILITY

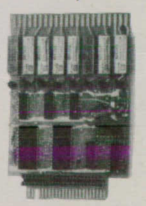
IN STOCK — SAME DAY SHIPMENT

BYTE-BACK'S BB-1 CONTROL MODULE

$59.00 KIT *In Stock!*

WIRED and TESTED $69.

- **8 Independent Relays**
  (with LED status indicators)
- **8 Independent TTL Inputs**
  (with Schmitt trigger buffers)



- By using a single POKE command you can change and latch the status of each of the 8 relays.
- Your **ZX80/1** can read the status of all 8 inputs by the use of a single PEEK command.
- A comprehensive manual is included that has complete application details.

RS-232 Module $59.95

Allows you to connect ZX81 to all RS-232 printers & terminals.

ALL MODULES CARRY 90-DAY WARRANTY

TRY BYTE-BACK MODULES FOR 10 DAYS WITH NO OBLIGATION

Remember with: BYTE-BACK modules you are NOT limited to using only one module at a time!

☐ M-64 Kit	$119.95	☐ M-16 Kit	$59.95
☐ M-64 Wired and Tested	$129.95	☐ M-16 Wired and Tested	$69.95
☐ M-64 Blank PC Board	$19.95	☐ M-16 Blank PC Board	$19.95
☐ BB-1 Kit and Manual	$59	☐ Modem Kit	$99.95
☐ BB-1 Wired and Tested and Manual	$69	☐ Modem Wired and Tested	$119.95
☐ BB-1 Blank PC Board and Manual	$29	☐ RS-232 Module	$59.95

Shipping and Handling $4.95

ORDER PHONE (803) 532-5812 **Bill My** ☐ Visa ☐ MasterCard

Exp. Date ____________ Card No. ____________________

Name ______________________________

Address ______________________________

City/State/Zip ______________________________

Mail To: **BYTE-BACK CO. • Rt. 3, Box 147 • Brodie Rd. • Leesville, S.C. 29070**

VISA

CHECKS

MasterCard

BYTE-BACK CO.
RT. 3, BOX 147 • BRODIE RD.
LEESVILLE, S.C. 29070
Ph. (803) 532-5812

perceptions

David B. Ornstein

The ZX81 Parser (Part 2)

In the last issue I began a discussion of the ZX81's parser which is a complex and tricky combination of software techniques. In this issue I will continue the discussion by detailing the main parser routines.

Syntax Checking vs. Execution

When you enter a line into the ZX81, the parser is called twice to evaluate it. The first time is when you hit ENTER after you have just typed a line into the system. This call instructs the parser to check the syntactic correctness of the line entered. If the line is in error, an S cursor is placed in the line at the appropriate position. The parser then returns to the keyboard routine to permit the user to correct his error. The process by which any syntax error is shown to the user is simple: an RST 8 instruction is executed. This instruction is followed by a byte which specifies the "REPORT NUMBER". The RST 8 begins execution by loading X-PTR with the value of CH-ADD, the system's interpreter pointer (see the *ZX81 BASIC Programming Manual,* chapter 28).

If there are no syntax errors, the parser returns, and the line is converted into its internal representation which is shown in Figure 1.

At this point, one of two actions will occur. If a line number was specified, the internal representation will be stored in the program area. Note that, if the text of the line is null (i.e., there is no text after the line number), and if a line number was given, the old version of the line, if any exists, will be deleted. If no line number was given, the parser will be called again. This time, however, it will execute the line.

Clearly, some method must exist to tell the parser whether it should check-syntax or execute the line passed to it. A bit (bit 7) in the FLAGS system variable is allocated for this use. If the parser is to execute the line, the bit will be set. Otherwise (i.e., syntax-check only), it will be reset. Thus the designation EXEC/SYNTAX. A routine SYNTAX-Z (0DA6) is used to check this bit. This routine will return Z true if just-checking is specified. It returns Z false (i.e., NZ) if execution is to occur.

To clarify the use of this flag, let me give an example. Suppose the parser is passed the following line:

PLOT 5,10 A

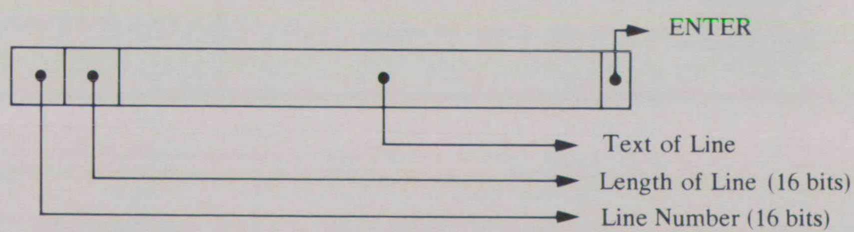
The code that will be executed is chosen by the parser. This process is detailed in "The Parser Body" below.

It must be made clear that the process of fetching arguments and checking syntax cannot be thought of as disjointed. In the case of the PLOT command, as is the case with almost all other commands which take arguments, all arguments are passed to Section B on the "calculator stack." (See "Understanding Floating Point Arithmetic: The CALCULATOR Language," by Ian Logan, *SYNC* 2:2.) As we will see in the next section, the routines used to fetch arguments check syntax, implicitly, as they move CH-ADD through the line—or rather the routines used to check syntax and fetch arguments, implicitly, as they move CH-ADD through the line.

The Parser Body

The ZX81's parser is table-driven. This means that all syntax checking and execution are directed by a series of tables.

Figure 1: The internal representation of a line in Basic.



Listing 1: The Parser Code.

```
P-OFFSET:       .BYTE $8B,$8D,$2D,$7F,$81,$49,$75
                .BYTE $5F,$40,$42,$2B,$17,$1F,$37
                .BYTE $52,$45,$ØF,$6D,$2B,$44,$2D
                .BYTE $5A,$3B,$4C,$45,$ØD,$52,$5A
                .BYTE $4D,$15,$6A
```

The parser code is given in Listing 1. There are two entry points into the parser: LINE-SCAN and LINE-RUN. When called via LINE-SCAN, the parser sets EXEC/$\overline{\text{SYNTAX}}$ to SYNTAX (i.e., 0), and calls E-LINE-NO. This subroutine is used to check the line number specified. If the line number is outside the bounds of a legal line number, a REPORT "C" is issued. If the line number is valid, LINE-SCAN falls through to LINE-RUN.

As you probably know, the ZX81 will accept an expression as the argument to an INPUT statement. The parser is used to evaluate this expression. The first action performed by the parser (entered via LINE-RUN) is to check whether an INPUT statement is being executed. This condition is specified by bit 5 of FLAGX. If an INPUT statement is currently being processed, one of the two actions will occur. If the first character is a STOP, and the EXEC/$\overline{\text{SYNTAX}}$ flag=EXEC, the parser will exit by using RST 8 to signal a REPORT "D". If the first character is not a STOP, then the parser jumps to INPUT-REP (see Class-2). If an INPUT statement is not being executed, a jump is made to LINE-NULL.

The parser, continuing execution at LINE-NULL, now checks to see if the line is null. If it is, the parser returns immediately. If the line is non-null, the parser checks to be sure that the first character in the line is a command. Once again, if it is not, REPORT "C" will be issued. Assuming that the first character in the line is a command, the parser continues by calculating the address of the command's entry in the parameter table. This is done by using an offset table (0C29). The offset table and the paramater table are shown in Listings 2 and 3, respectively.

Reviewing the parameter table, you will see that each command entry specifies the format of a particular command. For example, PLOT (0C98) is shown to be comprsised of a CLASS-6 item, a comma (,), and another CLASS-6 item. After finding the currently executing command in the table, the parser falls through into a loop (GET-PARAM), which is responsible for fetching successive parameter-bytes from the command's entry.

After fetching a byte, GET-PARAM checks its value. If the byte's value is greater than 0Bh, the GET-PARAM loop jumps to the SEPERATOR routine which checks the character in the line against the specified parameter-byte (i.e., or=, etc). If a match is found to exit, SEPERATOR returns to the beginning of the GET-PARAM loop. If the characters did not match, REPORT "C" is issued via RST 8.

Listing 2: The Offset Table.

```
P-LET:          .BYTE $01,$14,$02      ;Class 1
                                       ; " "
                                       ; Class 2
P-GOTO:         .BYTE $06,$00          ; Class 6
                .BYTE $81,$0E          ; Class 0
                                       ; Addr-$0E81
P-IF:           .BYTE $06,$DE,$05      ; Class 6
                .BYTE $AB,$0           ; THEN
                                       ; Class 5
                                       ;  Addr-$0DAB
P-GOSUB:        .BYTE $06,$00          ; Class 6
                .BYTE $B5,$0E          ; Addr-$0EB5
P-STOP:         .BYTE $00,$DC,$0C      ; Class 0
                                       ; Addr-$0CDC
P-RETURN:       .BYTE $00,$D8,$0E      ; Class 0
                                       ; Addr-$0ED8
P-FOR:          .BYTE $04,$14,$06      ; Class 4
                .BYTE $DF,$06,$05      ; " "
                .BYTE $B9,$0D          ; Class 6
                                       ; TO
                                       ; Class 5
                                       ; Addr-$0DB9
P-NEXT:         .BYTE $04,$00          ; Class 4
                .BYTE $2E,$0E          ; Class 0
                                       ; Addr-$0E2E
P-PRINT:        .BYTE $05,$CF,$0A      ; Class 5
                                       ; Addr-$0ACF
P-INPUT:        .BYTE $01,$00          ; Class 1
                .BYTE $E9,$0E          ; Class 0
                                       ; Addr-$0EE9
P-DIM:          .BYTE $05,$09,$14      ; Class 5
                                       ; Addr-$1409
P-REM:          ,BYTE $05,$6A,$0D      ; Class 5
                                       ; Addr-$0D6A
P-NEW:          .BYTE $00,$C3,$03      ; Class 0
                                       ; Addr-$03C3
P-RUN:          .BYTE $03,$AF,$0E      ; Class 3
                                       ; Addr-$0EAF
P-LIST:         .BYTE $03,$30,$07      ; Class 3
                                       ; Addr-$0730
P-POKE:         .BYTE $06,$1A,$06      ; Class 6
                .BYTE $00,$92,$0E      ; ","
                                       ; Class 6
                                       ; Class 0
                                       ; Addr-$0E92
```

```
                CALL  NZ,STK-FETCH
                LD   HL,FLAGX
                OR  (HL)
                LD  (HL),A
                EX  DE,HL
SET-STRLN:      LD  (STRLEN),BC
                LD  (DEST),HL
REM:            RET
CLASS-2:        POP  BC
                LD  A,(FLAGS)
INPUT-REP:      PUSH AF
                CALL SCANNING
                POP AF
                LD  BC,$1321
                LD  D,(FLAGS)
                XOR D
                AND $40
```

If, on the other hand, the parameter-byte's value is less than 0B, the parser uses another offset table (0D16) to find the address of the class handler. The class handler is then called. When it finishes execution, it will (probably) return to the beginning of the GET-PARAM loop.

Class Handlers

A class handler is a routine used to parse out a particular type of variable (i.e., non-constant) parameter from a line. These items include, but are not limited to, variable names and expressions. As class handlers perform extremely diversified functions, each will be presented separately with a description of the effects of its execution.

CLASS-6

CLASS-6 is the routine used to parse out an integral expression. CLASS-6 uses a ROM routine known as SCANNING which is a general expression parsing routine. Although a discussion of expression evaluation techniques is beyond the scope of this article, it suffices to say that the SCANNING leaves the result of its evaluation on the calculator stack. It also sets bit 6 of FLAGS to indicate whether the value parsed was of numeric or string type. When control is returned to CLASS-6, after its call to SCANNING, it checks bit 6 of FLAGS. If a string argument was specified, then REPORT "C" is given. If a numeric argument was supplied, the CLASS-6 returns.

CLASS-1

CLASS-1 is called to parse out an assignable variable name. This procedure begins by calling LOOK-VARS, a routine to pull a variable name out of the source line. LOOK-VARS returns a pointer to the variable's record in the VARS file. If the variable is not found, LOOK-VARS creates it, and returns a pointer to the created record. CLASS-1 stores the returned pointer and other pertinent information in system variables for later use. It then returns.

CLASS-4

CLASS-4 is a routine used for a purpose similar to that of CLASS-1. It is used to parse out the variable name specified as the argument to a FOR or NEXT command. It calls LOOK-VARS and checks the type bits returned in the C register. It checks to be sure that both bits 5 and 6 are set in the type byte. If they are not set, REPORT "C" is given. If the variable name was valid (i.e., a single-character numeric scalar), CLASS-4 will jump into the middle of the CLASS-1 routine which will then proceed by storing the information on the variable in the appropriate system variables.

Listing 2 (continued)

```
               JR   NZ,REPORT-C
               BIT  7,D
               JR   NZ,CLASS-END
               JRB  CHECK-2

CLASS-4:       CALL LOOK-VARS
               PUSH AF
               LD   A,C
               OR   $9F
               INC  A
               JR   NZ,REPORT-C
               POP  AF
               JR   CLASS-4-2

CLASS-6:       CALL SCANNING
               BIT  6,(FLAGS)
               RET  NZ

REPORT-C:      RST  8
               .BYTE $ØB

NO-TO-STK:     JR NZ,CLASS-6
               CALL SYNTAX-Z
               RET Z
               RST  28                     ; Call the
                                             calculator
               .BYTE $A0                   ; Stk-zero
               .BYTE $34                   ; End-calc
               RET

SYNTAX-Z:      BIT 7,(FLAGS)
               RET
               LD   HL,C-OFFSET
               LD    B,$ØØ
               ADD HL,BC
               LD C,(HL)
               ADD HL,BC
               PUSH HL
               RET

SEPERATOR:     RST 18
               CP   C
               JR   NZ,REPORT-C2
               RST 20
               RET

C-OFFSET:      .BYTE $17,$25,$53,$ØF,$6B,$13,$76

CHECK-END      CALL SYNTAX-Z
               RET NZ
               POP BC
CHECK-2:       LD   A,(HL)
               CP   NEWLINE
               RET Z
REPORT-C2:     JR   REPORT-C

CLASS-3:       CP   NEWLINE
               CALL NO-TO-STK

CLASS-0:       CP   A

CLASS-5:       POP  BC
               CALL Z,CHECK-END
               EX   DE,HL
               LD   HL,(T_ADDR)
               LD   C,(HL)
```

AARDVARK — THE ADVENTURE PLACE

ADVENTURES FOR OSI, TRS-80, TRS-80 COLOR, SINCLAIR, PET, VIC-20

ADVENTURES — Adventures are a unique form of computer game. They let you spend 30 to 70 hours exploring and conquering a world you have never seen before. There is little or no luck in Adventuring. The rewards are for creative thinking, courage, and wise gambling — not fast reflexes.

In Adventuring, the computer speaks and listens to plain English. No prior knowledge of computers, special controls, or games is required so everyone enjoys them—even people who do not like computers.

Except for Quest, itself unique among Adventure games, Adventures are non-graphic. Adventures are more like a novel than a comic book or arcade game. It is like reading a particular exciting book where you are the main character.

All of the Adventures in this ad are in Basic. They are full featured, fully plotted adventures that will take a minimum of thirty hours (in several sittings) to play.

Adventuring requires 16k on Sinclair, TRS-80, and TRS-80 Color. They require 8k on OSI and 13k on VIC-20. Sinclair requires extended BASIC.

TREK ADVENTURE by Bob Retelle — This one takes place aboard a familiar starship and is a must for trekkies. The problem is a familiar one — The ship is in a "decaying orbit" (the Captain never could learn to park!) and the engines are out (You would think that in all those years, they would have learned to build some that didn't die once a week). Your options are to start the engine, save the ship, get off the ship, or die. Good Luck.

Authors note to players — I wrote this one with a concordance in hand. It is very accurate — and a lot of fun. It was nice to wander around the ship instead of watching it on T.V.

CIRCLE WORLD by Bob Anderson — The Alien culture has built a huge world in the shape of a ring circling their sun. They left behind some strange creatures and a lot of advanced technology. Unfortunately, the world is headed for destruction and it is your job to save it before it plunges into the sun!

Editors note to players — In keeping with the large scale of Circle World, the author wrote a very large adventure. It has a lot of rooms and a lot of objects in them. It is a very convoluted, very complex adventure. One of our largest. Not available on OSI.

HAUNTED HOUSE by Bob Anderson — This one is for the kids. The house has ghosts, goblins, vampires and treasures — and problems designed for the 8 to 13 year old. This is a real adventure and does require some thinking and problem solving — but only for kids.

Authors note to players—This one was fun to write. The vocabulary and characters were designed for younger players and lots of things happen when they give the computer commands. This one teaches logical thought, mapping skills, and creativity while keeping their interest.

DERELICT by Rodger Olsen and Bob Anderson — For Wealth and Glory, you have to ransack a thousand year old space ship. You'll have to learn to speak their language and operate the machinery they left behind. The hardest problem of all is to live through it.

Authors note to players — This adventure is the new winner in the "Toughest Adventure at Aardvark Sweepstakes". Our most difficult problem in writing the adventure was to keep it logical and realistic. There are no irrational traps and sudden senseless deaths in Derelict. This ship was designed to be perfectly safe for its' builders. It just happens to be deadly to alien invaders like you.

NUCLEAR SUB by Bob Retelle — You start at the bottom of the ocean in a wrecked Nuclear Sub. There is literally no way to go but up. Save the ship, raise her, or get out of her before she blows or start WWIII.

Editors note to players—This was actually plotted by Rodger Olsen, Bob Retelle, and someone you don't know — Three of the nastiest minds in adventure writing. It is devious, wicked, and kills you often. The TRS-80 Color version has nice sound and special effects.

EARTHQUAKE by Bob Anderson and Rodger Olsen — A second kids adventure. You are trapped in a shopping center during an earthquake. There is a way out, but you need help. To save yourself, you have to be a hero and save others first.

Authors note to players — This one feels good. Not only is it designed for the younger set (see note on Haunted House), but it also plays nicely. Instead of killing, you have to save lives to win this one. The player must help others first if he/she is to survive — I like that.

PYRAMID by Rodger Olsen — This is one of our toughest Adventures. Average time through the Pyramid is 50 to 70 hours. The old boys who built this Pyramid did not mean for it to be ransacked by people like you.

Authors note to players — This is a very entertaining and very tough adventure. I left clues everywhere but came up with some ingenous problems. This one has captivated people so much that I get calls daily from as far away as New Zealand and France from bleary eyed people who are stuck in the Pyramid and desperate for more clues.

QUEST by Bob Retelle and Rodger Olsen — THIS IS DIFFERENT FROM ALL THE OTHER GAMES OF ADVENTURE!!!! It is played on a computer generated map of Alesia. You lead a small band of adventurers on a mission to conquer the Citadel of Moorlock. You have to build an army and then arm and feed them by combat, bargaining, exploration of ruins and temples, and outright banditry. The game takes 2 to 5 hours to play and is different each time. The TRS-80 Color version has nice visual effects and sound. Not available on OSI. This is the most popular game we have ever published.

MARS by Rodger Olsen — Your ship crashed on the Red Planet and you have to get home. You will have to explore a Martian city, repair your ship and deal with possibly hostile aliens to get home again.

Authors note to players — This is highly recommended as a first adventure. It is in no way simple—playing time normally runs from 30 to 50 hours — but it is constructed in a more "open" manner to let you try out adventuring and get used to the game before you hit the really tough problems.

ADVENTURE WRITING/DEATHSHIP by Rodger Olsen — This is a data sheet showing how we do it. It is about 14 pages of detailed instructions how to write your own adventures. It contains the entire text of Deathship. Data sheet - $3.95. NOTE: Owners of OSI, TRS-80, TRS-80 Color, and Vic 20 computers can also get Deathship on tape for an additional $5.00.

PRICE AND AVAILABILITY:

All adventures are $14.95 on tape except Earthquake and Haunted House which are $9.95. Disk versions are available on OSI and TRS-80 Color for $2.00 additional.

Please specify system on all orders

ALSO FROM AARDVARK — This is only a partial list of what we carry. We have a lot of other games (particularly for the TRS-80 Color and OSI), business programs, blank tapes and disks and hardware. Send $1.00 for our complete catalog.

AARDVARK - 80
2352 S. Commerce, Walled Lake, MI 48088
(313) 669-3110

Phone Orders Accepted 8:00 a.m. to 4:00 p.m. EST. Mon.-Fri.

TRS-80 COLOR SINCLAIR OSI VIC-20

Listing 2 (continued)

```
                INC HL
                LD  B,(HL)
                EX  DE,HL
CLASS-END:      PUSH BC
                RET

CLASS-1:        CALL LOOK-VARS
CLASS-4-2:      LD  (FLAGX),$00
                JR  NC,SET-STK
                SET  1,(FLAGX)
                JR NZ,SET-STRLN
REPORT-2:       RST 8
                .BYTE $01
SET-STK:        CALL Z,STK-VAR
                BIT  6,(FLAGS)
                JR  NZ,SET-STRLN
                XOR A
                CALL SYNTAX-Z
LINE-SCAN:      LD  (FLAGS),01        ; Set up SYNTAX-
                                        checking only
                CALL E-LINE-NO
LINE-RUN:       CALL SET-MEM
                LD   HL,ERR_NR
                LD  (HL),$FF
                LD   HL,FLAGX
                BIT 5,(HL)
                JR  Z,LINE-NULL
                CP  "STOP"            ; ie E3 hex
                LD   A,(HL)
                JP  NZ,INPUT-REP
                CALL SYNTAX-Z
                RET  Z

                RST 8                 ; Error "D"
                .BYTE $ØC

STOP:           RST 8                 ; Give REPORT "9"
                .BYTE $Ø8

LINE-NULL:      RST 18
                LD   B,$00
                CP   NEWLINE
                RET  Z

                LD   C,A
                RST  20
                LD  A,C
                SUB  $E1
                JR  C,REPORT-C2

                LD C,A
                LD  HL,P-OFFSET
                ADD HL,BC
                LD  C,(HL)
                ADD HL,BC
                JR  GET-PARAM

SCAN-LOOP:      LD  HL,(T_ADDR)
GET-PARAM:      LD  A,(HL)
                INC HL
                LD  (T_ADDR),HL
                LD  BC,SCAN-LOOP
                PUSH BC
                LD C,A
                CP  $ØB
                JR  NC,SEPERATOR
```

CLASS-2

The CLASS-2 routine is responsible for parsing out the value assigned to a variable (previously specified by CLASS-1) in a LET or INPUT statement. The expression is evaluated by SCANNING. The type of the expression evaluated is then checked against the type of the variable (name) stored by CLASS-1. If the types are not equal, then REPORT "C" is issued. If, on the other hand, the types do mix, a jump is made to the LET routine at 1321h.

CLASS-5

IF, PRINT, LPRINT, FOR, REM, LOAD, DIM, and SAVE all have CLASS-5 as their last parameter. The CLASS-5 byte (05h) is followed by two bytes which form an address. The address specified is that of the handler for that command. The command handler is the routine that actually performs the action specified by the command (i.e., placing a dot on the screen, in the case of a PLOT command). By reviewing the code for CLASS-5, you will see that, after POPing its return address (probably 0CF4h), its gets the next two bytes from the command's entry in the parameter table. It combines these bytes and jumps to the address specified by the resulting word.

Before executing the jump, a CALL is made, conditionally (if the Z-flag is set) to CHECK-END. This routine checks to be sure that the next character in the line specifies an end-of-line (i.e., a NEWLINE). If it does not, REPORT "C" is given. If the character is a NEWLINE, a return is made.

CLASS-0

CLASS-0 is simply a CP A instruction. This instruction is commonly used by Z80 programmers to set the Z-flag. After executing this instruction, CLASS-0 falls through to CLASS-5. CLASS-5 will then CALL CHECK-END, as the Z-flag is set.

CLASS-3

CLASS-3 handles the argument specified after a RAND, LIST, LLIST, or RUN command. This element can be either a NEWLINE or an integral expression. If a NEWLINE is given without any expression, a default of 0 is assumed, and pushed onto the calculator stack. If a number was specified, CLASS-6 is CALLed to parse the expression. After pushing the appropriate number onto the calculator stack, CLASS-6 returns to CLASS-3. CLASS-3 then falls through to CLASS-0.

In the next issue I will discuss the command handlers themselves.

Once again, I wish to extend my utmost gratitude to Ian Logan for his hard work in disassembling the 8K ROM. All labels in this article come from the result of his work: *The Sinclair ZX81 ROM Disassembly*, Part A.

Until next time, same relativistic time period, same non-Euclidian universe.

Listing 3: The Parameter Table.

```
P-RAND:      .BYTE $03,$6C,$0E    ; Class 3
                                  ; Addr-$0E6C
P-LOAD:      .BYTE $05,$40,$03    ; Class 5
                                  ; Addr-$0340
P-SAVE:      .BYTE $05,$F6,$02    ; Class 5
                                  ; Addr-$02F6
P-CONT:      .BYTE $00,$7C,$0E    ; Class 0
                                  ; Addr-$0E7C
P-CLEAR:     .BYTE $00,$9A,$14    ; Class 0
                                  ; Addr-$149A
P-CLS:       .BYTE $00,$2A,$0A    ; Class 0
                                  ; Addr-$0A2A
P-PLOT:      .BYTE $06,$1A,$06    ; Class 6
             .BYTE $00,$AF,$0B    ; ","
                                  ; Class 6
                                  ; Class 0
                                  ; Addr-$0BAF
P-UNPLOT:    .BYTE $06,$1A,$06    ; Class 6
             .BYTE $00,$AF,$0B    ; ","
                                  ; Class 6
                                  ; Class 0
                                  ; Addr-$0BAF
P-SCROLL:    .BYTE $00,$0E,$0C    ; Class 0
                                  ; Addr-$0C0E
P-PAUSE:     .BYTE $06,$00,$32    ; Class 6
             .BYTE $0F            ; Class 0
                                  ; Addr-$ØF32
P-SLOW:      .BYTE $00,$2B,$0F    ; Class 0
                                  ; Addr-$0F2B
P-FAST:      .BYTE $00,$23,$0F    ; Class 0
                                  ; Addr-$0F23
P-COPY:      .BYTE $00,$69,$08    ; Class 0
                                  ; Addr-$0869
P-LPRINT:    .BYTE $05,$CB,$0A    ; Class 5
                                  ;Addr-$0ACB
P-LLIST:     .BYTE $03,$2C,$07    ; Class 3
                                  ; Addr-$072C
```

Kitchen Sync

Alan Groupe, Michael Tardiff, and Ivan Zatkovich

Two to the What?

Recently, a cousin from Lincolnwood, Illinois, came out to visit. Along with him, he brought a couple of "extra credit" problems given him by his eighth grade science teacher.

One of the problems was to calculate 2 to the 420th power. The problem came with the following restriction: "You can't use a computer." Since no commonly available computer can store a number this large with perfect accuracy (remember, floating point numbers are only approximations of a number), this restriction seemed unnecessary.

You have probably already guessed, though, that with some thought and a few tricks, you can solve this problem using a computer. It should also come as no surprise that we are going to show you how to do it on your ZX80/81.

It does seem to us, however, that any eighth grader who could figure out how to solve this problem on a computer is well on the way to becoming a clever problem-solver who can make good use of available tools, and, therefore, deserves the extra credit. Lincolnwood School District, take note.

To solve this problem on a ZX80/81 or on any computer, you first have to understand how to do it by hand.

Unless you are a child prodigy, you cannot handle a number as big as 2 to the 420th either. The best way to deal with things too big to handle all at once, of course, is to break them down into "bite-size" chunks. In this case, the most obvious method probably would be to treat each digit separately. You could take 2, multiply it by 2, take that answer and multiply it by 2 again, and so on, 419 times. When the numbers begin getting too large to multiply in your head, you would move from right to left, multiplying a digit by 2, adding in any carry from the previous digit, and carrying into the next, if necessary. Just like in school, right?

While the ZX80/81 cannot handle numbers as large as 2 to the 420th, it can handle numbers as big as 32767 (decimal). Rather than doing the calculations one digit at a time, as you would do by hand, it is a simple matter to have the ZX80/81 multiply four digits at a time. Listing 1 gives the program for calculating 2 to the 420th. This was written on the ZX80 4K ROM, but it works also on the ZX81 if you make this change:

180 PRINT A(I);"##";

Since the answer to our problem is 127 decimal digits long (How do we know? Trust us.), we can store it in an array of 32 integers. This is done in line 10. Lines 20-50 initialize our answer to 1 (2 to the zero-ith power). Lines 60 through 160 are the main loop of the program and multiply A by 2 420 times. C is the carry from one "digit grouping" to the next and is cleared in line 70. (There is no carry, of course, into the rightmost digit.)

Lines 80 and 90 set up a backwards FOR-NEXT loop going from 32 down to 1, covering each digit group from right to left. We could have numbered the digit groups in the other direction, but then we would have needed a reverse loop to print out the result, so it did not make much difference which way we went. Line 100 multiplies the current digit group by 2 and adds in the carry from the previous digit group. Line 110 clears the carry, since we do not as yet know whether there is to be a carry out of the current digit group. Line 120 skips over the carry processing if the current digit group is within range (0-9999). Lines 130 and 140 do the carry processing by bringing the digit group back within range and setting the carry into the next group to 1. You should satisfy yourself that this is correct.

Line 150 closes the loop at line 80 and proceeds to the next digit group. Line 160 closes the loop at line 60 and proceeds to the next doubling. Calculations are now finished, and lines 170 through 190 print out the result.

Note that the format of the output is rather sloppy. The reason for this is that some of the digit groups may be less than 1000, and in order for the result to be printed together, program lines would have to be added to print these groups with leading zeros. While this is not particularly difficult, it would tend to make the program more complicated. If you have sufficient RAM, the inspiration, and a compulsion for neatness, see if you can alter our program to "prettyprint."

When run, the program grinds away for a little over four minutes before returning with the answer. Compared to pencil and paper, that is not bad. But consider this: on the first iteration of the main loop (lines 60-160) the variable A(32), which is originally 1, is multiplied by 2, giving 2. But A(1) through A(31), while being equal to 0, are still multiplied by 2. On the second iteration this is also true. In fact, most of the four minutes are spent multiplying 0 by 2!

This provides us with a good example off how careful planning can make a program run faster. If you think through a problem ahead of time, you can cut out the pointless work and free your ZX80/81 to do the real work of calculating or searching or whatever. Sure enough, there is a simple way to "optimize" our program to eliminate the wasted work of multiplying 0 by 2.

If you keep a pointer (call it "P") to the most significant (leftmost) non-zero digit, and stop the right-to-left multiplication loop (lines 80-150) at that point, you avoid the unnecessary multiplications. Remember, though, to move this pointer to the left when you carry out of that digit (in this case, the digit to the left will now be non-zero).

If that is a bit tough to follow the first time through, add the lines in Listing 2 for 4K ROM or Listing 3 for 8K ROM. Then trace through the program using pencil and paper. You will see how the unnecessary work is avoided.

Software Publishers—Analyze the NEED!

The TIMEX and SINCLAIR ZX systems are cassette-based. Everything depends on delivering software product the user can LOAD very readily, without difficulty. In publishing software for these machines, reputation and repeat orders depend on good programming AND ready-loading product.

Software Publishers—Qualify the SOURCE!

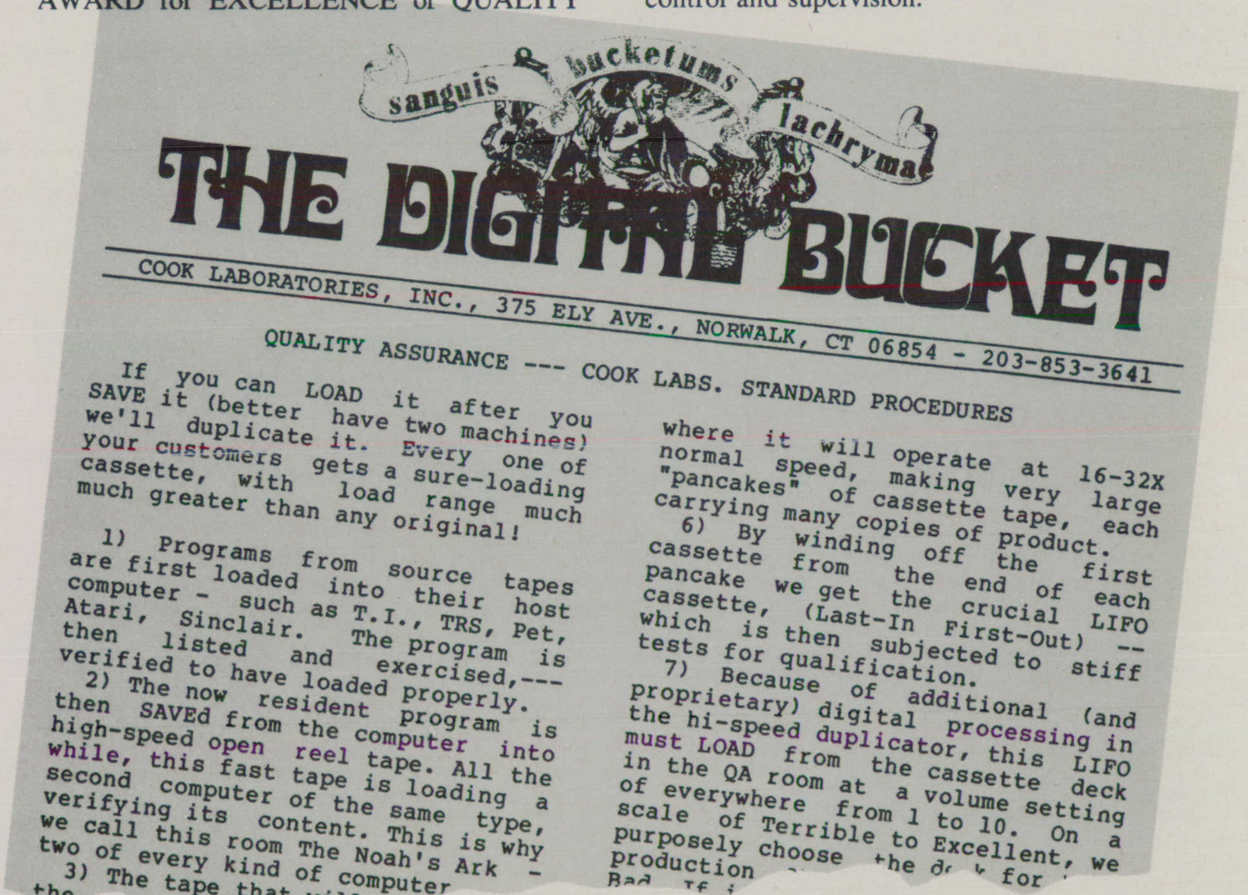
COOK LABS is the product source that DOES duplicate digitally — not audio-analog.

COOK LABS has been making digital cassettes for serious software publishers for many years. Why?

COOK LABS is the only duplicating source with Texas Instruments' CERTIFICATE of AWARD for EXCELLENCE of QUALITY (1981). Users don't have problems with T.I. cassette software. They needn't with yours.

SINCE 1952 Emory Cook's name has been internationally associated with leadership in electronics design.

COOK LABS operates under his engineering control and supervision.



sanguis bucketums lachryma

THE DIGITAL BUCKET

COOK LABORATORIES, INC., 375 ELY AVE., NORWALK, CT 06854 - 203-853-3641

QUALITY ASSURANCE --- COOK LABS. STANDARD PROCEDURES

If you can LOAD it after you SAVE it (better have two machines) we'll duplicate it. Every one of your customers gets a sure-loading cassette, with load range much much greater than any original!

1) Programs from source tapes are first loaded into their host computer - such as T.I., TRS, Pet, Atari, Sinclair. The program is then listed and exercised,--- verified to have loaded properly.

2) The now resident program is then SAVEd from the computer into high-speed open reel tape. All the while, this fast tape is loading a second computer of the same type, verifying its content. This is why we call this room The Noah's Ark - two of every kind of computer

3) The tape that will ...

... where it will operate at 16-32X normal speed, making very large "pancakes" of cassette tape, each carrying many copies of product.

6) By winding off the first cassette from the end of each pancake we get the crucial LIFO cassette, (Last-In First-Out) -- which is then subjected to stiff tests for qualification.

7) Because of additional (and proprietary) digital processing in the hi-speed duplicator, this LIFO must LOAD from the cassette deck in the QA room at a volume setting of everywhere from 1 to 10. On a scale of Terrible to Excellent, we purposely choose the deck for production ...

SOFTWARE PUBLISHERS — PHONE, WRITE, USE COUPON . . . GET INFORMATION NOW!

COOK LABORATORIES, INC.
P.O. Box 529
Norwalk, CT 06856 (Phone 203-853-3641)

- ☐ Inform me about COOK services for:
  - ☐ Software duplication.
  - ☐ Procurement of labels, inserts, etc.
  - ☐ Fulfillment, including fulfillment of individual orders.
- ☐ Send free booklet: "Cassettes — An Effective Software Medium."

NAME__________

COMPANY__________

ADDRESS__________

__________ ZIP__________

PHONE (OPTIONAL)__________

SYNC Program Listings

Readers should note the following conventions used in the program listings in this issue:

— The number symbol is used in PRINT statements to show necessary spaces.

"A"— The underline is used in PRINT statements to indicate graphics. Press the key designated in whatever way your machine requires to get the graphic.

INPUT — In PRINT statements when a word or token is underlined enter the word from the keyboard rather than type it in letter by letter. This is a memory saving technique.

After typing in a program line, you must put the line into your program. On the ZX80 and ZX81 with 8K ROM this is done by hitting the NEWLINE key. On the ZX81 the same thing is done by hitting ENTER. The words are used interchangeably.

A number followed by a b is binary; a number followed by h is hex.

Line 55 initializes our new pointer to one significant digit group. This is because A(32), or rather, A(P), is the only digit group with anything in it. Line 80 has been changed to loop through only the significant digit groups. You should verify that this is correct.

Lines 143 through 147 have been added to the carry processing section. They will only be executed when there is a carry out of a digit group. Line 143 checks to see if this is a carry out of the currently most significant digit group (the one P points to). If not, this is simply a normal carry, and carry processing is done. Otherwise, lines 145 and 147 move the pointer one digit group to the left and set that digit group to 1 (the carry out of the previous group).

With this modification in place, the program takes just under three minutes, an improvement of over 35 percent.

It should be apparent that you can use this same technique to calculate other "big" numbers. If you really want to impress your friends, amaze yourself, and give your ZX80/81 a workout, try calculating that wonderful constant, pi, to a hundred places or so. You will have to solve a few problems along the way, but you have two months until the next issue of *SYNC* comes in the mail...right?

By the way, in case you did not have your ZX80/81 with you as you read this column (perish the thought!), the answer to our problem is (drum roll, please): 2, 707, 685, 248, 164, 858, 261, 307, 045, 101, 702, 230, 179, 137, 145, 581, 421, 695, 874, 189, 921, 465, 443, 966, 120, 903, 931, 272, 499, 975, 005, 961, 073, 806, 735, 733 604, 454, 495, 675, 614, 232, 576.

Listing 1: 2 to the 420th (4K ROM).

```
 10 DIM A(32)
 20 FOR I=1 TO 32
 30 LET A(I)=0
 40 NEXT I
 50 LET A(32)=1
 60 FOR N=1 TO 420
 70 LET C=0
 80 FOR I=1 TO 32
 90 LET I1=33-I
100 LET A(I1)=A(I1)*2+C
110 LET C=0
120 IF A(I1)<10000 THEN GOTO 15
0
130 LET A(I1)=A(I1)-10000
140 LET C=1
150 NEXT I
160 NEXT N
170 FOR I=1 TO 32
180 PRINT A(I),
190 NEXT I
```

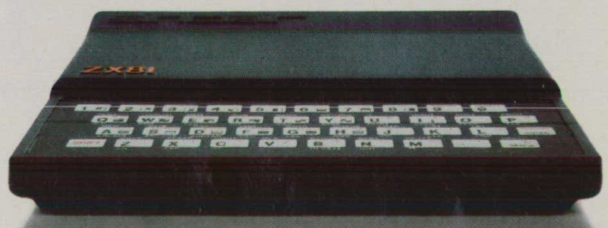
Listing 2: Eliminating 0*2 (4K ROM).

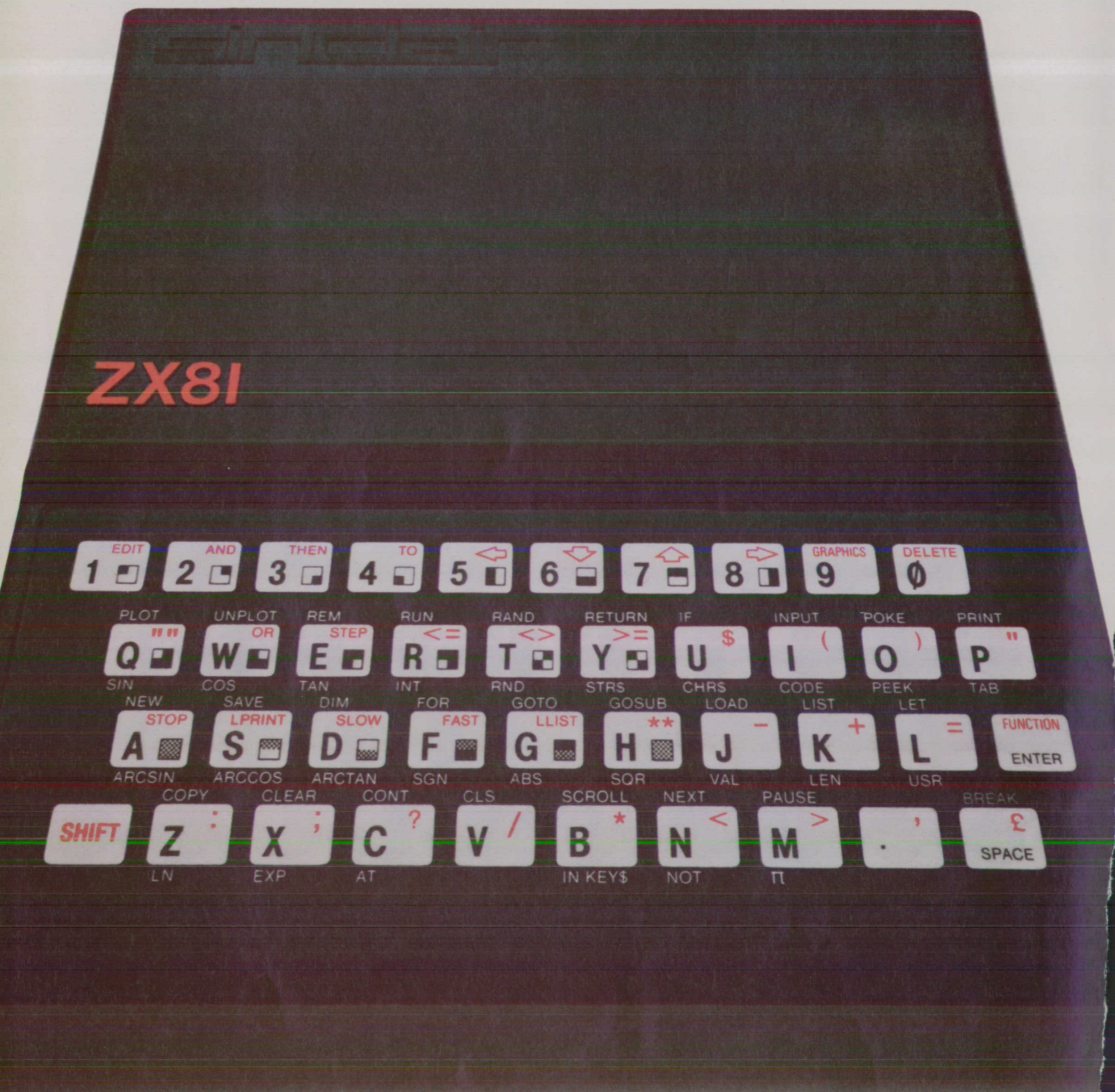
```
 55 LET P=32
 80 FOR I=1 TO 32-P+1
143 IF NOT I1=P THEN GOTO 150
145 LET P=P-1
147 LET A(P)=1
```

Listing 3: Eliminating 0*2 (8K ROM).

```
 55 LET P=32
 80 FOR I=1 TO 32-P+1
143 IF I1<>P THEN GOTO 150
145 LET P=P-1
147 LET A(P)=1
```

Finally you can afford to satisfy your lust for power.




ZX81
EDIT 1
AND 2
THEN 3
TO 4
5
6
7
8
GRAPHICS 9
DELETE Ø
PLOT Q "" SIN
UNPLOT W OR COS
REM E STEP TAN
RUN R <= INT
RAND T <> RND
RETURN Y >= STR$
IF U $ CHR$
INPUT I (CODE
POKE O) PEEK
PRINT P " TAB
NEW A STOP ARCSIN
SAVE S LPRINT ARCCOS
DIM D SLOW ARCTAN
FOR F FAST SGN
GOTO G LLIST ABS
GOSUB H ** SQR
LOAD J - VAL
LIST K + LEN
LET L = USR
FUNCTION ENTER
SHIFT
COPY Z : LN
CLEAR X ; EXP
CONT C ? AT
CLS V /
SCROLL B * IN KEY$
NEXT N < NOT
PAUSE M > π
. ,
BREAK £ SPACE

For $99.95 you can have a full powered personal computer.

Most people know by now that the ZX81 from Sinclair Research is the lowest priced personal computer in the world.

But serious programmers are looking for more than a low price. They're looking for true computer power. And that's where the ZX81 surprises a lot of people.

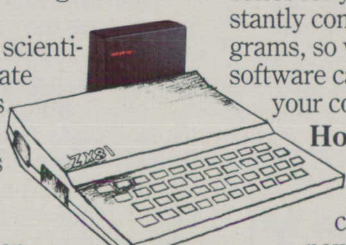
Just look at the keyboard and you'll get some idea of the ZX81 's power. It has more than 60 BASIC commands, 20 graphic symbols, and complete mathematical functions. And there's even more power that you can't see.

Sinclair programs are available

A breakthrough in personal computers. The ZX81 offers features found only on computers costing two or three times as much.

Just look at what you get:

- Continuous display, including moving graphics
- Multi-dimensional string and numerical arrays
- Mathematical and scientific functions accurate to 8 decimal places
- Unique one-touch entry of key words like PRINT, RUN and LIST
- Automatic syntax error detection and easy editing
- Randomize function useful for both games and serious applications
- Built-in interface for ZX Printer
- 1K of memory expandable to 16K
- A comprehensive programming guide and operating manual



16K Memory Module

The ZX81 is also very convenient to use. It hooks up to any television set to produce a clear 32-column by 24-line display. And you can use a regular cassette recorder to store and recall programs by name.

What you get. When you order your ZX81, you get everything you need to start programming.

It comes with connectors for your TV and cassette recorder, an AC adaptor, and a free programming guide and operating manual that completely documents the capabilities of the ZX81.

Options and add-ons. Like any full-powered computer, the ZX81 can be expanded and upgraded.

Its 1K memory can be expanded to over 16K just by plugging the Sinclair Memory Module onto the back of the unit. The cost is only $49.95.

Sinclair has also published pre-recorded programs on cassettes for your ZX81. We're constantly coming out with new programs, so we'll send you our latest software catalog when you order your computer.

How did we do it? The question most often asked about the ZX81 is, "How can so much computer power cost so little money?"

The answer is that Sinclair Research simply took a different approach. Our only goal was to make programming power as affordable as possible. So we developed a radical new design that cuts costs dramatically without cutting computer power. For example, our unique Master Chip replaces as many as 18 chips used in other personal computers.

The success of the ZX81 speaks for itself. It is now the fastest-selling personal computer in the world. And we stand behind our product. If anything goes wrong in the first 90 days, we'll repair or replace your unit free of charge. Even after that, you can take advantage of our national service-by-mail facilities for a minimum fee.

Order now and try it out for 10 days. Simply send the coupon along with a check or money order. For faster delivery, call our toll-free number and use your MasterCard or VISA.

You have 10 days to try out the ZX81. If it isn't all we say it is, just send it back and we'll refund your money.

Why wait any longer? With the Sinclair ZX81, you can finally afford to have the computer power you've always wanted.

Call toll free 800-543-3000. Ask for operator #509. In Ohio call: 800-582-1364; in Canada call: 513-729-4300. Ask for operator #509. Phones open 24 hours a day, 7 days a week. Have your MasterCard or VISA ready.

Free guide to programming

These numbers are for orders only. If you just want information, please write: Sinclair Research Ltd., 2 Sinclair Plaza, Nashua, NH 03061.

To order call toll free: 800-543-3000.

AD CODE 07SY — MAIL TO: Sinclair Research Ltd., One Sinclair Plaza, Nashua, NH 03061.

	PRICE*	QTY.	AMOUNT
ZX81	$99.95		
16K Memory Module	$49.95		
Shipping and Handling	$4.95		$4.95
*U.S. dollars	TOTAL		

Name

Address

City State Zip

Sinclair technology is also available in Timex/Sinclair computers under a license from Sinclair Research Ltd.

sinclair

Understanding Your ZX81 ROM

Thomas L. Keeney

Understanding Your ZX81 ROM by Dr. Ian Logan. Melbourne House Ltd., Glebe Cottage, Station Rd., Cheddington, Leighton Buzzard, Bedfordshire LU7 7NA, U.K. 162 pp.; paperback. Available directly from the author for $18 (airmail). In U.S. $19.95 from: Gladstone, Heuristics, Sinclair Place, Softsync.

SYNC readers will recognize Dr. Logan as a major ZX80/81 expert whose writings are well worthwhile. *Understanding Your ZX81 ROM* is no exception. The book is written for a serious beginner who has a fairly good knowledge of Basic and wishes to take advantage of Z80 machine code execution speed. It supplements, but does not replace, the ZX81 manual which must be used as a reference. In many ways this book is an extension of Dr. Logan's article "An Introduction to Machine Code" in *SYNC* 1:6.

The book falls logically into five parts. The first is a brief review of the Z80 CPU, its registers, and its instruction format. Next a quick introduction to the internal arithmetic and number base manipulations required is included.

Thomas L. Keeney, 9629 Dortmund, Huntsville, Al 35803.

The second part is a discussion of the Z80 machine code instruction set grouped by function. Examples are drawn liberally from the 8K ROM. Dr. Logan includes 26 additional examples in the form of simple Basic programs which can be entered and executed in the 1K memory.

A detailed examination of the relevant parts of the 8K ROM makes up the third part. The discussion is limited, however, to those routines which the user can reasonably access and use. Again, simple Basic programs detail the manner in which ROM based subroutines can be applied to a program. A simple USR-(address) will not work with most of them. ROM functions for report generation, character printing, keyboard input, and display generation are discussed.

The fourth part might properly be called "Getting Started in Machine Code." Dr. Logan introduces machine code programming by treating it as an extension to Basic. His technique of writing Basic programs that mimic machine code is unique to my experience. This appears to be such a good teaching method that it should become widespread. Of course, the technique is possible only because the Sinclair Basic can accept variables of any length. Such concepts as flowcharting and program structure help get the beginner started. Dr. Logan leads the programmer to think in terms of the special needs of the CPU. Finally, he discusses the integration of the Basic and machine code into one program. Demonstration programs illustrate the whole process.

The fifth part consists of appendixes. Complete listings of the more important 8K ROM routines are given as well as the usual tables of machine code language instructions, decimal-hexadecimal conversions, and keycodes.

Obviously, the book contains a great deal of information, and you would probably want to keep it as a reference. However, the book can be made easier to use with three improvements.

First, the book clearly needs an index and a revision of the physical arrangement of some of the contents. For example, the Z80 instructions set descriptions and the 8K ROM examples are separated from the short programs illustrating their use. This forces the reader to flip back and forth between references. I spend a lot of time with my fingers stuck in several pages to keep track of the relevant subject references while trying to apply the information to a particular programming problem.

Second, since the 8K ROM is decimal oriented, the inclusion of decimal equivalents for the hexadecimal numbers would remove some frustration for the beginner. Granted that a book on machine language needs some hex code information and examples, but the extensive use of hexadecimal leads to some problems. For example, the table of machine code instructions in Appendix ii is useless as a quick reference until the reader himself enters the decimal equivalents.

Third, a more durable type of binding that will allow the book to lie open and flat so that the reader can use both hands to enter program listings is needed. The pages are starting to fall out of mine, and I will soon ring bind it myself.

In spite of these three complaints, the fact that I have worn out Dr. Logan's book is an indication of how useful I have found it to be. The fact that I intend to repair it is an indication of how valuable I think it will continue to be. It contains a tremendous amount of information and is well worth the cost. *Understanding Your ZX81 ROM* would be a valuable addition to most ZX80/81 libraries, and for the serious programmer it should be high on the list.

Sinclair ZX Spectrum

David Tebbutt

Well, he's done it again, hasn't he? Uncle Clive has gone and shown the world how to produce a decent colour personal computer at the sort of price only he can conceive. Two versions of the Spectrum are available — 16k and 48k at £125 and £175 respectively, *including* VAT. If you want to upgrade your 16k machine to 48k later on, it will cost you £60. At the moment, the machines are available only by mail order. The Spectrum (ZX Spectrum to be precise) offers colour, high resolution graphics and sound and, at the price, it has just got to be the best value for money around.

Like most computers at this level, the Spectrum plugs into the domestic television and uses a normal cassette recorder. The ZX printer can be attached and, with a little modification, ZX81 programs will run happily on the Spectrum. A £50 miniature disk drive (the ZX Microdrive), communications facilities and an RS232 interface will be announced later on.

Hardware

The Spectrum measures just 233x144x 30mm and weighs in at 520 grams excluding the separate power supply and cables. It looks extremely elegant and, unlike its predecessors, it has keys that actually press down.

You'll not be surprised to learn that there are hardly any components inside the machine: 14 chips, a UHF modulator, a piezo-electric 'speaker' and an assortment of capacitors, resistors, diodes, crystals and a coil make up the complement. I swear that some of my crystal sets had more in them. All this is mounted on a single board and, looking underneath this production prototype, I notice that there's not a single patch. The only odd thing about it is that there's a big blob of green plasticine stuck around the coil. The coil on the review machine does whistle a bit but I understand that production machines come with suitably lacquered coils to eliminate this problem.

Reprinted with permission of *Personal Computer World*, 14 Rathbone Place, London W1P 1DE, England.

A hefty edge connector at the back brings out just about every signal you could wish to have. This is used for printers, communications and disk drive connections. Inside there are two spare sockets which accommodate each end of the 32k memory expansion board. This is a great improvement on the ZX81 memory expansion which tended to drop off the back of the machine at the least provocation. Talking of sockets (well I was, just now), every chip except the ULA is socketed. The reason the ULA isn't is because it gets darned hot — putting it on the PCB allows the heat to dissipate better.

The keyboard comprises a one-piece grey rubber moulding mounted over a pressure-sensitive membrane. The keys poke up through holes in a black metal plate and I must confess the feel is more that of a calculator than a typewriter. Most keytops have three symbols on them and, in addition, most of them have another two associated inscriptions printed on the metal surround. If you're anything like me you'll find yourself reading the whole keyboard each time you want to find a function. You do get used to it after a while: in my case it took a couple of days. I found that red symbols on grey keytops are quite difficult to read and, thinking my eyesight might be going, I showed the machine to a number of friends, all of whom had the same difficulty. I showed it to my 11-year-old and he thought it was just fine, though.

A power supply is included in the price, so there's not a lot of point risking one of your own and blowing the Spectrum up. The two cassette leads terminate in 3.5mm jack plugs so be sure that they work with your recorder before you embark on any major programs. It took me four or five tries before I found the right volume setting on my tape recorder. Once this was found, though, program loading presented no problems.

I tried the Spectrum on three televisions and the results matched the quality of the sets used. The display comprises 24 lines of 32 characters with the bottom two lines reserved for messages and entries. The display can also be regarded as 176 x 256 resolution for graphics work. High resolution graphics work is best done in two colours as you will see in the Firmware section of this review. The screen, border and individual characters can each take on one of eight colours and, in addition to this, characters can be bright or flashing. Other screen attributes like inverse and overprinting relate to the whole screen. More on these later.

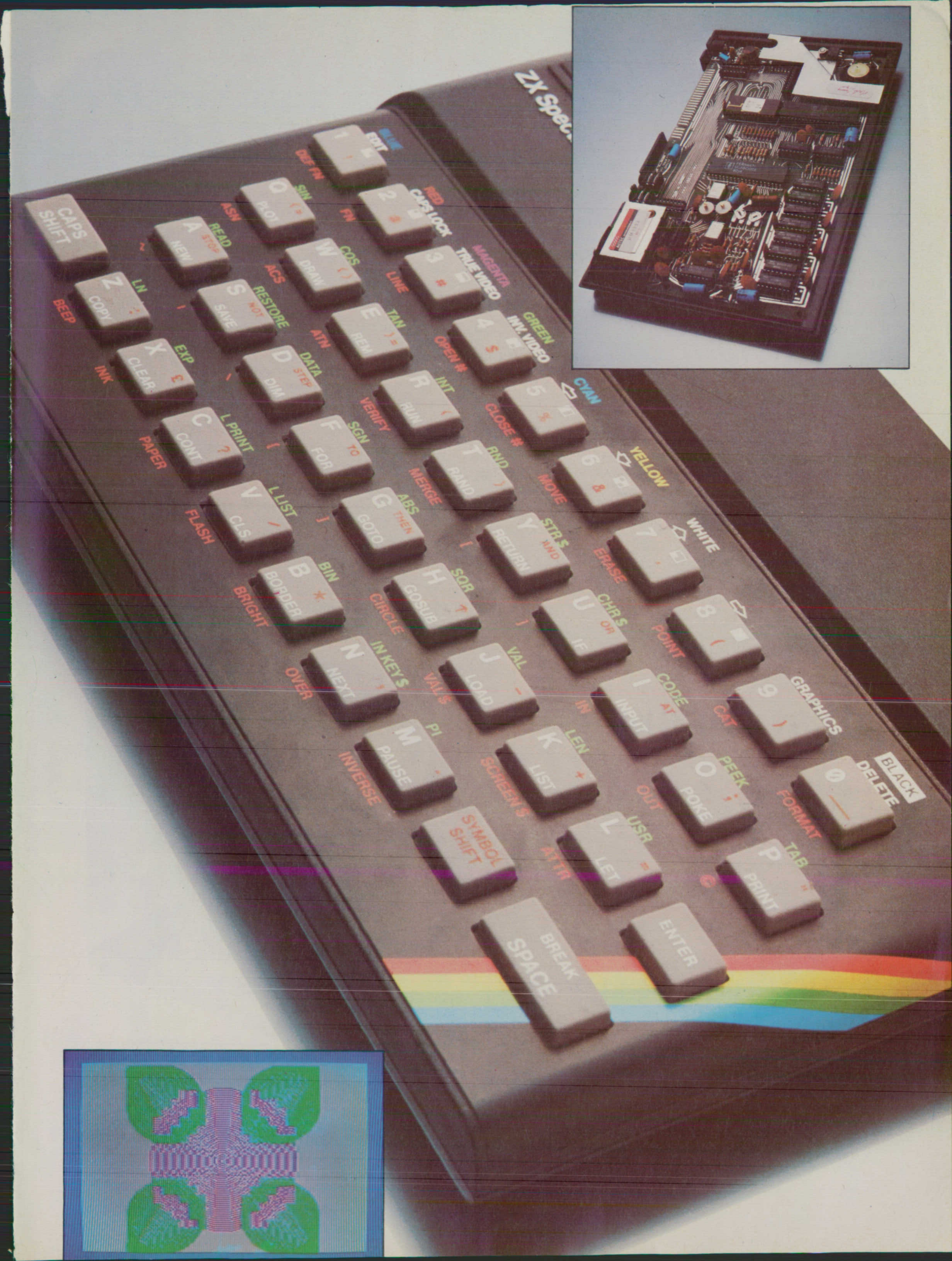
The single channel BEEP facility is about what you'd expect from a piezo-electric speaker. It does sound slightly better amplified from the cassette port but it's still pretty awful. A couple of octaves around middle C aren't bad; but the other eight are best used for sound effects. At the high end they warble and at the low end they grate — BEEP is a refreshingly honest description.

Really, there's not a lot more to say about the hardware. It is a very professional job; looks smart, works well and manages to squeeze 191 legends on to just 40 keys!

Firmware

Here's a new section for *PCW* Benchtests. All the software on the review machine was in the ROM chip which also contained the character set. This time Sinclair has gone for a basic ASCII set (upper and lower case) with the addition of both built-in and user-defined graphics characters. Outside of the range SPACE to QUOTES (32 to 126), many of the codes have special values relating to Spectrum keys and functions. For example, you'll find a copyright symbol key. (Now why didn't anyone else think of that?) You can define up to 21 characters of your own.

Two screen tables are maintained in memory — one for the displayed characters themselves and the other for the attributes which describe how they're to be displayed. These attributes can be tested from within a Basic program. The character colour is referred to as INK while the background colour is called PAPER. Isn't that sensible? Each character can have its own value for INK, PAPER, FLASHING, BRIGHTNESS, INVERSE and OVER. The last two should be explained: INVERSE simply means that the dots which form the character are printed in the PAPER colour while the PAPER is printed in the INK colour. OVER is special: it allows you to merge a new character with the one already at the screen position. The rules are that two INKs or two PAPERs print PAPER otherwise it prints INK. This means that you have a neat way of removing the last thing



CAPS SHIFT
SYMBOL SHIFT
BREAK SPACE
ENTER
GRAPHICS
BLACK
DELETE
WHITE
YELLOW
CYAN
GREEN
MAGENTA
RED
BLUE
TRUE VIDEO
INV. VIDEO
CAPS LOCK
EDIT

printed and restoring what was there before it.

By now you have probably realised why it is best to stick to two colours when doing graphics work. Since the colour of the INK and PAPER relates to a whole character position, then each time a new colour graphics point is set, all other set points within the boundary of that character are set to the new colour. This makes for a very curious effect to say the least.

Mathematical accuracy is to 9½ decimal digits and a fairly full range of mathematical functions is accessible from the keyboard. While on the subject of keyboards, this one has a built-in software 'click', an upper-case lock key and automatically repeating keys. Like the ZX80 and ZX81 before it, the Spectrum makes great use of single stroke keyword entries. In fact, I think every standard function and command is obtainable in this way. You'll even find things like >= and <= occupying their own pieces of grey rubber.

Basic

The Spectrum comes with a very useful version of Basic. It will be quite familiar to anyone who is used to the Microsoft types of Basic and a doddle to learn for those new to the language.

Rather than go through all the features and functions of the language, I have summarised them in a separate box. Here, I'll just comment on the unusual and interesting aspects of this particular implementation. Unlike some Basics, it is a teeny bit strict about things like using LET before assigning a value to a variable name or putting GOTO after a THEN. My view is that this is all jolly good discipline and it is more than compensated for by the fact that Spectrum pops in all those spaces which make programs so much easier to read. Of course, once you've found your way round the keyboard, the single stroke keyword entry is a joy. (I've got a feeling I said that in my last two ZX reviews.)

SAVEing and LOADing cassette tapes gives plenty of scope on this machine. You can save a program normally, you can save it so that execution starts automatically when it is reloaded, you can save arrays, you can save particular chunks of memory and if you want to keep a pretty picture you've created then you can use the SCREEN$ option to save that too. All saved programs can be verified after saving. The screen save can't be verified because the display is changed during the verify program and it would not then match that held on tape. The LOAD command can, of course, handle any tape created by SAVE. The MERGE command allows you to merge a program on tape with one already in memory. Program lines which are duplicated are overwritten while all others are suitably interleaved.

The graphics facilities are great fun. You can draw straight lines, curves and circles on the 176x256 pixel (PICture ELement, or dot) window. Position 0,0 is at the bottom left-hand corner of the screen. You can define up to 21 graphics characters of your own which is a superb feature if you're into writing your own Space Invader or Pack-Man games. I had a lot of fun drawing and animating little people on the screen. The nice thing is that you can do all this sort of thing without leaving Basic. A BIN (binary) notation has been introduced which allows you to define numbers as a series of 0s and 1s — just the ticket for designing funny characters. Each character comprises eight lines of eight points, so a succession of eight BIN numbers is all you need to define such a character. Another use for user-defined graphics is to squeeze some extra colours out of the machine. If you lay out the 64 pixels like a chess board and choose suitable INK and PAPER colours then you can get some interesting effects. Most of them will be awful but persevere — some will be good.

You won't be surprised to learn that line drawing and circle plotting are achieved using the DRAW and CIRCLE commands. A PLOT command allows you to plot single points. POINT enables you to find out whether a parti-

ZX80—ZX81 HARDWARE

ZX80—ZX81. Hardware.
TRI. Tape Recorder Interface unit available for those with **LOADING** problems. Self contained with leads and instructions this is a cheaper solution than a more powerful cassette machine!
£12. + 50 p. p&p in UK.

V.U.I. Video Amplifier Unit.
This self contained unit allows the ZX81 to drive a standard video monitor, gives a much better picture than on a TV set. Connections take a few minutes. Complete with leads or instructions. £12. + 50 p. p&p in UK.

D. BRUCE ELECTRONICS
THE BEACON
BLACKHALL ROCKS
CLEVELAND TS27 4BH
ENGLAND TEL. 0783 863612.

NEW! ZX81 16K RAM NEW!

BLACKJACK*MASTER by R.A.Foley
Up to 7 Players with "casino" options like Double Down, Split Pairs, Surrender, Sit Out, and more! Don't settle for less.

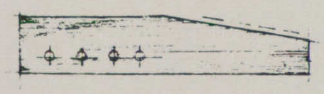
FOOTBALL FORECASTER by G.Smith
Get this one QUICK as the data base is for the 82-83 season. Specify COLLEGE or PRO version. Money Order gets rush delivery.

REVERSI by Peter Ulrich
2 Players, 1 against the 81, OR make it play itself. Fast MC moves + optional RATED list of possible moves. "A#1" version!

$19.95 ea.+ 2.50 for the order. Each ZETA Program is shipped on a certified cassette tape with informative booklet. Catalog sent with order or write:

ZETA SOFTWARE
PO BOX 3522/GREENVILLE, SC 29608

WSM ENTERTAINMENT has:



CONVERT YOUR ZX 81 TO A FULL SIZE COMPUTER WITH A STANDARD KEYBOARD

With a little soldering and a few parts you can convert you zx to a full size standard keyboard with a lock-shift key. (A must for fast, trouble free programming.)

PLANS INCLUDE:

Parts list, source of materials, directions, and working drawings for a solid walnut case to house computer and keyboard........$4.00

Conversion plans with solid Walnut case ready for keyboard and ZX (Only a few left in stock) $ 35.-

W S M ENTERTAINMENT

P.O. BOX 5367

North Hollywood, Ca.
91616-5367

$

slots

WHEN YOU ORDER WSM's "SLOTS" YOU'LL SEE GRAPHICS GENERATED SLOT MACHINE WITH LIVE ACTION LEVER MOVEMENT, WINNINGS DISPLAYED, AND ALL THE FUN OF A REAL LAS VEGAS SLOT MACHINE....

only you can CHANGE the payoff rate!

FUN FOR KIDS & ADULTS ALIKE!!!

16K only (cassette).......$ 5.-

Shipping and handling.....$ 2.-

PROGRAMS 25¢*

Twenty beginner programs for 1 K ZX 81 machines. All are printed and include instructions.

BASIC GAMES GRAFSTARTER©

ELECTRONIC CALENDAR FUNFONT©

...And many more! A great starting library of ZX basic language.

$ 5.-

* *in lots of twenty, of course.*

For information on any of the following items, send us a self addressed legal size , stamped envelope and we'll rush you more information.

SPEED EDITING ON THE ZX 81 *a guide to editing shortcuts with ZX 81's*

CARTOONS *with a computer? Yes! now you can create live action cartoons, or just characters*

MOVIE STUNTMAN *you experience the high falls and car crashes - not to mention the agony of DEFEAT!*

TEACH YOUR TOT *a parent assisted program of learning to share with your child*

THE COMPANY *THE ultimate CIA adventure game of espionage and terrorism!*

ADULTS ONLY! *the guessing games for wives and other lovers. Comes in a plain brown wrapper. (Rated PG)* cular pixel is set. You always DRAW from where Spectrum thinks you are on the screen. For example, a command DRAW 10,10,pi would draw a semi-circle ending up 10 places to the right and 10 above the current position. A fraction of pi would provide a different arc while zero, or no third argument (DRAW 10, 10) would draw a straight line. The curve can be drawn on either side of the centre line by making the third argument a positive or negative number.

The CIRCLE command uses three arguments: x-axis, y-axis and radius. Remember, the OVER command can be used to erase something already drawn. I used this feature in conjunction with DRAW, PLOT and CIRCLE to create cartoon effects. OVER is also useful for embedding text in a drawing. When set on, the text merges with the existing lines in the drawing. When set off it prints the full 8x8 character, completely replacing anything already displayed at that position. Incidentally, SCREEN$ can be used to return details of the contents of a character position. Used in conjunction with the PRINT AT command, this could be a good way of making your program find a suitable place to print a sort of 'label' on a drawing. The AT allows you to define the row and column at which printing should start.

A few instructions I particularly noticed as I went though the manual were READ, DATA, RESTORE and VAL$. READ and DATA are old friends although I can't remember them being on previous ZX machines. Using the DATA command you can provide lists of information at the beginning of a program. Each READ instruction takes the next word from this list. RESTORE can be used to set the DATA pointer to any DATA statement. VAL$ baffles me — it strips the outside quotes from string expressions and returns the string value of the result. Perhaps some kind reader would care to suggest a worthwhile application for this feature.

Now let's have a look at our honestly named friend, BEEP. There's not a lot to tell, really, except that you can control both pitch and duration. Notes below middle C are represented by negative numbers, those above by positive. Twelve numbers make an octave. (If you look at a piano keyboard you'll find that there are seven white notes and five black notes per octave.) Middle C is zero. The duration is expressed in seconds or fractions of a second. As I mentioned earlier, the sound isn't brilliant but it has the saving grace of

Spectrum Basic

Functions

ABS	ACS	AND	ASN	ATN	ATTR	BIN	CHR$	CODE
COS	EXP	FN	IN	INKEY$	INT	LEN	LN	NOT
OR	PEEK	PI	POINT	RND	SCREEN$	SGN	SIN	SQR
STR$	TAN	USR	VAL	VAL$				

Operations

+ − * / = > < <= >= <>

Statements

BEEP	DATA	GO TO	LOAD. .DATA	OVER	RETURN
BORDER	DEF FN	IF. .THEN	LOAD. .CODE	PAPER	RUN
BRIGHT	DELETE	INK	LOAD. .SCREEN$	PAUSE	SAVE
CAT	DIM	INPUT	LPRINT	PLOT	SAVE. .LINE
CIRCLE	DRAW	INVERSE	MERGE	POKE	SAVE. .DATA
CLEAR	ERASE	LET	MOVE	PRINT	SAVE. .CODE
CLOSE	FLASH	LIST	NEW	RANDOMIZE	SAVE. .SCREEN$
CLS	FOR. .TO. .STEP		NEXT	READ	STOP
CONTINUE	FORMAT	LLIST	OPEN #	REM	VERIFY
COPY	GOSUB	LOAD	OUT	RESTORE	

Sinclair Place

SEPTEMBER'S FEATURE PRODUCT

Quicksilva Sound Board

- 4096 seperate tones on each of 3 channels
- Volume control of each channel
- White noise generator with pitch control
- Envelope shaping facilities
- Two 8 bit in/out ports

QS Sound Board **$49.50**

OTHER QUICKSILVA PRODUCTS

QS Motherboard and QS Connector

- Allows use of RAM pack and up to 2 add-on boards.
- 5V regulator cures overheating
- Connector and Motherboard plug into back of ZX
- Permits use of add-ons such as QS Sound Board, QS Hi-Res Graphics Controller, and QS Programmable Character Generator

QS Motherboard **$27.75**
QS Connector **$9.25**

QS Hi-Res Graphics Controller

- 256 (horizontal) by 192 (vertical)
- Has 2K of ROM and 6K of RAM — fully buffered
- No software required
- Easily connected to QS Motherboard

QS Hi-Res Bd **$159.95**

QS Programmable Character Generator

- Program up to 128 different characters at a time
- Characters are stored in 1K of RAM on the board
- Each character is built on an 8x8 square
- No charge is necessary to existing programs
- Operating cassette included

QS Character Bd **$49.50**

QS 3K RAM Boards

- Gives total of 4K of RAM
- Does not suffer from "amnesia" and "shrinkage"
- Plugs into rear port of ZX or QS Motherboard

QS 3K RAM **$34.75**

Software

- QS Harmony
  - A program designed to exploit the musical potential of the QS Sound board **$8.95**
- QS Asteroids
  - ZX-81 + 4K RAM minimum **$9.95**
- QS Invaders
  - ZX-81 + 8K RAM minimum **$9.95**
- QS Defenders
  - ZX-80 or 81 8K ROM + 4K RAM minimum **$9.95**
- All QS software is recorded twice on quality cassettes
- All are written in machine code for fast moving graphics
- The programs have received very good ratings

JP GIBBONS

Personal Banking System

- Single key operation
- Search by check number, description or amount
- File can be displayed, printed, added to, cancelled or amended
- Validation of entries
- Requires 16K and cassette

Personal Banking System **$19.95**

MELBOURNE HOUSE

Books

- Not only 30 programs for The Sinclair ZX81 .... **$14.95**
- Machine language made simple **19.95**
- ZX81 ROM Disassembly Part A & B **$14.95** each (2 volumes) — Dr. Ian Logan
- Understanding Your ZX81 ROM .... **19.95** Dr. Ian Logan
- The Sinclair ZX81 basic course **35.00**

Cassette Tapes

ZX Chess — 16K	ZX80-8K/ZX81	**24.50**
10 Exciting Programs — 1K	ZX81	**14.95**
Adventure A — 16K	ZX80-8K/XZ81	**19.50**
Adventure B — 16K	ZX80-8K/ZX81	**19.50**
Adventure C — 16K	ZX80-8K/ZX81	**19.50**
Spacetrek — 16K	ZX81	**14.95**
Directory/Record — 1K with listing	ZX80-8K/ZX81	**14.95**
Reversi — 1K	ZX80-8K/ZX81	**14.95**
Super Invasion — 1K	ZX80-8K/ZX81	**14.95**
Wallbusters — 1K	ZX80-8K/ZX81	**14.95**
ZX Galaxians — 16K	ZX81	**14.95**
ZX Bug Utility Program — 16K	ZX81	**24.50**

If you are a producer of **quality** hardware or software for Sinclair products and are interested in having a U.S. or Westcoast organization marketing your products we would like to hear from you.

NO HANDLING OR SHIPPING CHARGES.
Washington residents add 6.5% sales tax.

P.O. Box 2288, Redmond, Washington 98052

being fairly quiet (ComputerTowns please note). You can pick this sound up from the cassette ports if you so wish. I'd say these facilities are more likely to be used for sound effects than composing symphines.

That's really all I have to say about the Basic. It is a very good implementation for a machine of this size. A PAINT instruction would have been nice to fill in graphics shapes, but I think it would look a bit weird in multicolour mode with the colours changing at each character boundary. A routine to do this should be simple enough. I think the screen resolution is quite adequate for most personal users of the machine. In fact you can churn out some quite stunning effects using DRAW, PLOT, CIRCLE and the user-defined characters.

Before moving on to documentation, here's a list of the disk commands just to whet your appetite: CAT, CLOSE, DELETE, ERASE, FORMAT, MOVE, OPEN. CAT is probably short for Catalogue which lists the files on a disk. MOVE probably copies a file from one place to another. The others are self-exaplanatory.

Documentation

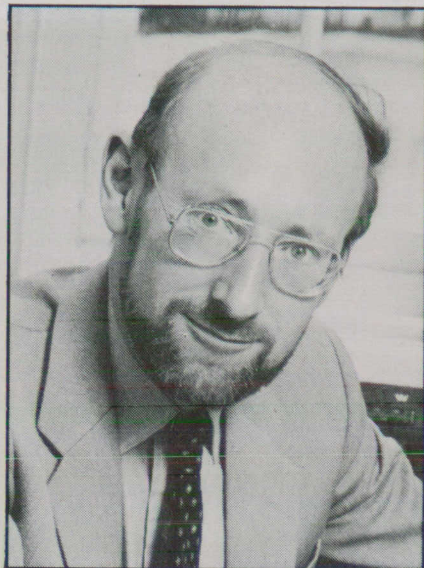
Two manuals come with the Spectrum — a thin but useful introduction for the complete novice and a thicker one which expiains things in depth. A lot of effort has been put into this latter manual. It is professionally presented and easy to read. Unfortunately, I was given a photocopy of the final proofs and it contained no index and no table of contents. I read the whole manual a couple of times before starting the review and I found it a real problem to find things that I knew were there somewhere. I must admit that the style wasn't to my liking; it's a little verbose and the individual chapters seem to lack structure. I also found the inevitable errors which might cause a beginner problems — things like a minus sign being printed instead of equals, for example. The manual certainly seems to cover everything, so if a table of contents and a comprehensive index are added you'll probably find it adequate. It's certainly an improvement on many manuals on the market.

Potential use

This is the first machine that I've reviewed since the Atari two summers ago that I would actually buy — in fact I will have probably ordered one by the time you read this. I would use it for fun, for fooling around with graphics and for programming in Z80 code. I would treat it as a hobby machine, a way of relaxing. My children have already become very interested in the graphics capability and I see this as a way of giving them a real understanding of mathematics. A Logo system on this at the right price would go down an absolute treat — if anyone out there thinks of doing it, I'd love to review it. Of course, there are those who want to learn to write programs. Once again, this is an excellent machine to cut your teeth on. I think that schools and

What about the '81?

It was no secret that 'Uncle' Clive was going to launch a knockout micro — he's put a bomb under the industry twice already, producing machines which brought computing power within everybody's reach at prices which drastically undercut the competition.



A slightly upmarket (by Sinclair standards) machine offering colour and sound and reasonable graphics at a price far, far below that of any equivalent machine was a logical step to take, especially in view of Sinclair's obvious annoyance at being left out of the BBC deal.

What is interesting, though, is that the Spectrum does not replace the ZX81, as the 81 did the 80 — it's an addition to the range and the ZX81 will continue in production. In fact, production of the 81 is to be *increased* to a target of 150,000 a month by the end of the year.

'The ZX81 will continue to be ideal for the person who wants the lowest possible entry cost into computing,' says Sinclair. And to prove the point, he's knocked £20 off the price of the 16k RAM pack.

At the moment Spectrum is available only by mail order and is only on sale in the UK — there are no plans yet to market it through retail outlets, as is done with the ZX81 through W H Smith, and export versions are not planned until the end of the year, with the USA being the first (and largest) market to get the new machine.

Peter Rodwell

homes have got to be the prime targets for Spectrum at the moment.

Later on when the disk drives appear, this may change. At a predicted £50 for a 100k drive, a lot of people who will have written the Spectrum off as a hobby machine will have to think again. Add to that a £20 combined RS232 and communications facility, and you could be talking about some very interesting and fairly sophisticated networks. At that stage, it becomes a very real prospect for schools looking for a fairly grown-up system, but one which can involve as many pupils as possible. At Sinclair's prices could we possibly be heading for the 'one on every desk' scenario painted by so many futurologists?

Until those disks arrive there is no great office potential for the Spectrum. Once they're on stream then it's probably just a question of appropriate software. Information management and Visicalc-type applications would seem to be the most likely and, because of the price of the television, they will probably be used with portable black and white machines. No doubt the dedicated will take their Spectrums (or is it Spectri?) home to plug into the colour TV. Most people will probably wait until Sinclair announces a flat screen colour television. The network idea could then be useful in offices for things like telephone directories, notice-boards and memos.

Prices (inc VAT)

16k Spectrum	£125
48k Spectrum	£175
32k Memory	£60
Microdrive disk	(estimated) £50
RS232 + Comms	(estimated) £20
ZX Printer	£59.95

Conclusions

Well, for the benefit of those who only read the first and last paragraphs of these reviews here are my conclusions: Clive Sinclair has produced a very good 16k personal computer which offers colour, high resolution graphics and limited sound for just £125. That represents very good value for money provided that this is the sort of machine you want. It is ideal for people who want to learn about computing and have a lot of fun while they're doing it. Given the right sort of graphics-based educational software, it can bring people very pleasurable ways of learning subjects such as mathematics and geography. Once the games programs start to appear, a lot of people will use it just for that, although it does seem a bit of a waste.

Later on, the provision of disk drives and communications facilities will make it an even more serious contender for the school markets and it will begin to creep into businesses. When the flat screen television appears then I suspect that the business interest will rise because the price will be far more appropriate. Bulletin boards, memos, telephone directories, spreadsheet calculation and information management seem to be the most likely applications.

The 'proper' keyboard is a distinct improvement on its predecessors, but it still doesn't achieve — or try to achieve — the quality of an IBM. All the old regular Sinclair features are included — the single keyword entry and the automatic syntax checking as you enter each command, for example.

My verdict? The best value for money you can find today!

I would like to thank John Mathieson of Sinclair Research for so patiently answering my questions.

Benchmark timings

BM1	4.8	BM3	21.1	BM5	24.0	BM7	80.7
BM2	8.7	BM4	20.4	BM6	55.3	BM8	25.3

All timings in seconds.

Technical data

CPU:	3.5 MHz Z80A
Memory:	16k Dynamic RAM, 16k ROM, 32k expansion option
Keyboard:	40 keys rubber moulding. 183 functions + 8 colour labels. Auto-repeat
Screen:	Domestic colour television. 8 Colours or 6 grey shades.
Cassette:	Domestic recorder.
Disk drives:	To be announced. 100k per drive.
Ports:	To be announced. RS232 & comms.
Language:	Basic in ROM

Double Your Memory

J. Wayne Schneider

Ed.—**A WORD OF CAUTION:** *Any hardware project for your computer must be approached with extreme caution.* **SYNC** *cannot be responsible for any problem that may arise from attempting hardware projects. Obviously, any damage to your computer can be costly in time and money.*

The first few days of working (playing?) with a new Sinclair ZX81 are filled with anticipation, excitement, and, for the kit builder, triumph and pride. Seeing that little black K on the screen when the ZX81 is first powered up brings exclamations of joy from the new programmer and a questioning stare from uninitiated bystanders. Alas, all of this excitement soon dwindles into frustration with the first attempt to impress a sceptic viewer with a simple random graphics display. There is not enough RAM for even one full screen and a modest program.

Do not despair! There is a reasonably quick and almost painless solution. Although the ZX81 was designed to be inexpensive, a foresighted designer has provided the circuitry to use a 2K x 8 bit static RAM on the circuit board. The whole process involves removing the two 2114s at IC4a and IC4b, installing a 2K x 8 RAM at IC4, and adding a jumper at L2. If you are lucky enough to have a ZX81 with a 4118 RAM, you will not even have to change IC sockets. Simply remove the 4118, install the 4802 in its place, and change the jumper at L1 to L2. One warning for those with factory built computers: performing the procedure described here will *void your warranty.*

J. Wayne Schneider, 1601 South 450 West, Provo, UT 84601.

Preparations

You will need a clean, dry workspace at least two by four feet. The area should be well-lighted and protected from solder drippings. An old bath towel spread over a table makes an excellent surface on which to work, but make sure it is an expendable towel before you burn a hole in it with a hot soldering iron. Furthermore, it protects both the table and the ZX81.

You will also need the following:

1) A light, 15-25 watt soldering iron with a fine tip.

2) Fine gauge rosin core solder.

3) A pair of sharp wire cutters.

4) A Phillips screwdriver with a no. 1 point.

5) A 2K x 8 static RAM. These are manufactured by several companies. Mostek's 4802 is no longer being manufactured, but it is still available from many electronics supply stores. Mitsubishi makes an equivalent part that costs less, number M58725P. Hitachi's CMOS part, number 6116, which requires considerably less power, is your best bet if you are concerned about the ZX81 overheating. Others are the Toshiba 2016, the Fujitsu MB8128, and the NEC part 4016.

If your ZX81 has the two ICs labeled uPD2114LC, you will need to buy a 24 pin DIP socket for the new RAM or else two 12 pin SIP sockets. The DIP (dual inline pin), socket is structurally better, but the SIPs can be installed without removing the old 18 pin socket. You will need some solder wick to help with the desoldering of the old 18 pin DIP socket if you are going to add a 24 pin DIP socket.

New rubber feet or glue will be needed when you reattach the rubber feet to the back cover of the ZX81.

When you have all of the parts and about three hours to spare, you are ready to begin.

Operations

Get all of the parts and tools together. Disconnect the ZX81 from everything, especially from the power supply. Place it flat in front of you just as though you were going to type on it.

1) Remove the back cover.

Turn the ZX81 over so that it is positioned as shown in Figure 1. There are five screws to remove. Only two of them are visible. The other three are under the rubber feet at positions 1, 4, and 5. Carefully remove the three rubber feet and save them for later. If you are lucky, the sticky will come with them, and they can be stuck to a piece of wax paper for installation later. Now remove all five screws. Notice that screws 4 and 5 are shorter. Set these in a safe place. Gently lift or pry the back cover off. It should come easily without forcing.

2) Remove the printed circuit board.

Remove the two screws that hold the printed circuit board in place. Note their exact position for reassembly. Very carefully lift the circuit board and rotate it back towards you until it is face up. Be extra careful not to stress or crimp the flat tails connected to the keyboard. You may now disconnect the keyboard by pulling straight and steady on one tail at a time. Set aside the cover with the keyboard attached.

3) Remove the existing 1K RAM.

Turn the keyboard around 180 degrees so that the flat square metal heat sink is on the lower left as shown in Figure 2. Remove the two 18 pin ICs at IC4a and IC4b. These are the 2114 1K x 4 bit RAMs. If your ZX81 does not have these two ICs, then it will have only one 24 pin IC at IC4. Remove this IC. It is a 4118 1K x 8 bit RAM. Save the RAM where it will not be subject to extreme temperatures or static.

4) Install the new IC sockets.

If you are one of the lucky ones whose ZX81 came with a 4118 RAM, you can ignore this process. Otherwise, carry on. Place the new sockets in the lower 12 holes on each side of location IC4. If the SIPs do not fit easily into the holes or if you are using a DIP socket, you will have to desolder and remove the 18 pin DIP socket. Now turn the board over and solder the new socket in place. (Caution: never bring a hot soldering iron close to a chip or any other component. Be sure that all the chips have been removed.) Be careful with the solder. Use enough to fill the hole but not so much that it runs down the pins and shorts something out. Turn the board back over.

Figure 1: Bottom of the ZX81.

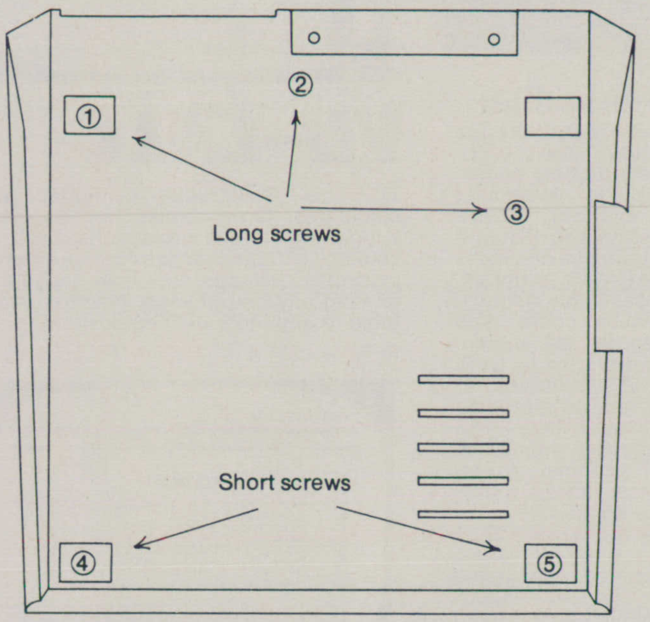
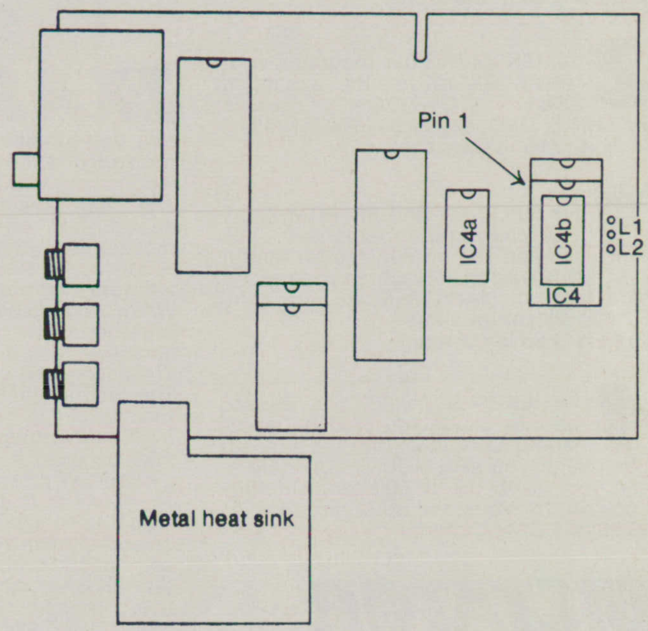


Figure 2: Board Orientation and IC Location.



5) Install a jumper at L2.

Those who have the 4118 RAM must first desolder the jumper at L1. Use a short piece of wire bent in a U shape to connect the solder pads at L2. Solder it in place. Be careful not to get too much solder.

6) Install the new 2K RAM IC.

Place the new 2K RAM in the new socket. It should be oriented with pin 1 to the upper left. Finding pin 1 may be a problem. The IC will probably have a dot etched or molded into the top surface adjacent to pin 1. If you cannot find a dot, there will be a notch in the center of one end. The notch should face away from you. If you had a 4118 RAM, be sure you are installing into the lower 24 holes. Nothing goes into the upper four holes. Push the socket into place. You may find it necessary to bend the pins inward to make it fit. After the IC is in place, inspect it to be sure that all the pins are in their holes and not bent underneath the IC.

7) Reassemble the ZX81.

Turn the printed circuit board 180 degrees so that the square metal heat sink is at the upper right. Place the cover beneath the circuit board. The keyboard tails should form a graceful loop back over the upper left corner and into their sockets. Be very careful when pushing these fragile tails into the sockets. If they are bent or torn, you will not have much success in getting the ZX81 to work. Rotate the circuit board back into place in the cover. Check the keyboard tails for positon. Screw the board in place with two of the short screws at the positions you noted in step 2. Replace the cover and screw it down using the short screws at locations 4 and 5. Do not use the long screws in the wrong place. You could damage the keyboard. Glue the rubber feet into place. When you install the rubber feet, place them to the side of the screw holes, and you will never have to go through that misery again.

Trouble

Connect your ZX81 to your TV again and plug in the power. If it works, treat yourself. If not, check the solder joints. You may have to resolder them. Do not resolder with the IC in the socket. Check the IC for orientation. Be sure all of the pins are in the holes. Did you forget the jumper? If the computer displays the inverse K but will not respond to the keyboard, check the tails. They should be seated straight in their sockets. The fault rarely lies within the RAM IC itself. However, that is a possibility. At that point you will need someone experienced in working with digital circuits to help you solve the problem.

Conclusion

Now that you are up and running, try the following command:

```
PRINT PEEK 16389*256-16384
```

That should display 2048, which is the amount of memory you now have available. A full screen requires 704 bytes of memory. The system variables require 125 bytes. With 1K of RAM, you are left with 195 bytes for a program. With 2K of RAM you are left with 1219 bytes; over six times as much program space. Of course, the ZX81 Basic does not really dedicate 704 bytes of memory unless your program fills the thirty-second position of every line. That will rarely happen, but now you can enter a reasonably long program and still have a full screen display if you want it.

ZX81 **SOFTWARE** On Cassette

GAMES PACKS

1 for 1K ZX81 & 8K ROM ZX80. Eight fantastic programs for the unexpanded ZX81, including DIGICLOCK, 9-LIVES, REACTION TEST, GOBBLER and PATTERNS.
$9.95 ($12.95 in Canada)

2 for 16K ZX81. Four programs written in BASIC for the expanded ZX81. PONTOON, FRUIT MACHINE, OXO, and BIO— RHYTHMS.
$9.95 ($12.95 in Canada)

3 for 16K ZX81 and 8K ROM ZX80. Two programs for expanded ZX81 to keep you entertained for hours! 3-D OXO is written in machine code and is hard to beat. MARS RESCUE is a compulsive adventure game.
$9.95 ($12.95 in Canada)

4 for 16K ZX81. ZOMBIES — escape as they chase you around Zombie Island. Lure them into the pits, but don't fall in yourself. MOUNT MAYHEM — can you reach the 20,000 foot summit? Look out for Yetis and other hazards!
$9.95 ($12.95 in Canada)

DICTATOR

Fantastic new adventure game for 16K (or greater) ZX81. You have just become 129th ruler of Ritimba with a single goal in mind: take full advantage of the situation for your own good. You have to deal with a handful of factions: unruly army, downtrodden peasants — but you have the secret police on your side.
$14.95 ($17.95 in Canada)

CONSTELLATION

Turn your ZX81 into a telescope with this amazing 16K program. Produces a simulation of the night sky as seen from any position on Earth at any chosen time this century. You may point your telescope in any direction, move it up, down, left or right, zoom in or out. Stars may be displayed by magnitude or constellation
$14.95 ($19.95 in Canada)

CHESS

A challenging chess programme, written in machine language, designed to operate in the ZX81 fast mode. ZX Chess allows you to select from 6 levels of play, choose either black or white, and enables castling and en passant moves. Unique "self-running" feature: you start the tape and when the chess board appears on the screen, start your game.
ZX CHESS! NOW $12.95

ZXAS Assembler

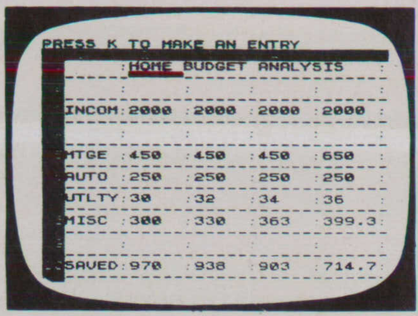
Now you can use the full power of the Z80 microprocessor without having to laboriously POKE in instruction codes. This full specification Z80 assembler assembles all the standard Zilog mnemonics, which are simply written into REM statements (more than one per line is allowed) within your BASIC program. When assembled, the assembly listings, together with assembled codes and adresses, are displayed on the screen. The assembled code is executed by USR. The program occupies 5K, is situated at the top of the memory, and is protected from overwriting. This means that ZXAS may be used in conjunction with ZXDB (see below), providing an extremely powerful machine code system normally only found on very expensive computers.
The program is available for both the ZX81 and the 8K ROM ZX80, and in both cases, the 16K RAM pack is required. Despite the low price, ZXAS is a FULL-SPECIFICATION assembler, and is a must for all serious ZX users. Full documentation on how to use the assembler (including a list of the mnemonics) is supplied.
$9.95 ($12.95 in Canada)

ZXDB Disassembler/ Debugger

The perfect complement to the ZXAS assembler, ZXDB is a complete combined machine code disassembler and debugging program. Like ZXAS, it is itself written in machine code for compactness, and may be used in conjunction with ZXAS, still leaving about 9K of memory for your own program.
Apart from the DIASSEMBLER, the program has features including SINGLE STEP, BLOCK SEARCH, TRANSFER AND FILL, HEX LOADER, REGISTER DISPLAY and more, all of which are executed by simple one key commands from the keyboard. All in all, an extremely powerful programming aid, well worth the money for the disassembler alone!
$9.95 ($12.95 in Canada)

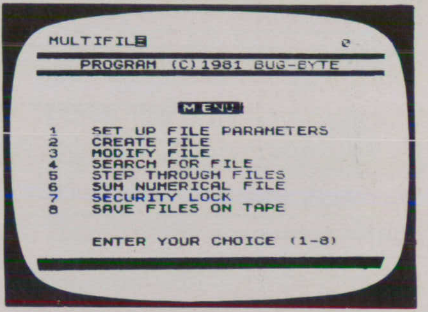
STAR TREK

The classic computer game in which you trek across the galaxy in search of Klingons to zap with your phasers and photon torpedoes. You have long and short range scanners to help you find them, Starbases to refuel your ship at and, of course, various witty comments from the crew. 16K.
$9.95 ($12.95 in Canada)



VU-CALC

VU-CALC. Constructs, generates, and calculates large tables for analysis, budget sheets and projections. Up to 26 columns of figures or data can be entered, plus user definable formulae capable of relating any one or more position in the table to any other defined position.
NOW $14.95



MULTIFILE

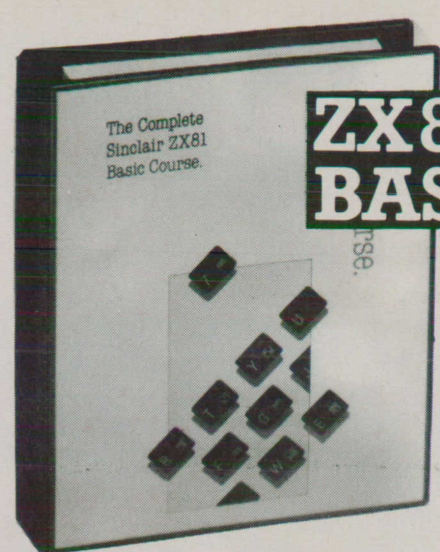
Data Storage System
An amazingly versatile multi-purpose filing system for the 16K ZX81. The program is menu-driven, and number, size and headings of files are user-definable. Both string and numerical files are catered for. Files may be created, modified, replaced, and searched, and are protected by an ingenious foolproof security system. Output to the ZX printer is also provided. The program comes on cassette, together with three quality data cassettes for file storage, and comprehensive documentation, describing a host of applications for both business and personal use. Supplied in an attractive storage case. If your ZX81 is bored with playing games, then this program will give it plenty to think about!
$29.95 ($39.95 in Canada)

New Multifile Plus for 64 K RAM $34.95

VIEWTEXT

A ten page information display system for the 16K ZX81. Can display both text and graphics in any sequence with variable speed. Many applications including shop window displays, educations, animation, etc.
$14.95 ($17.95 in Canada)

Gladstone Electronics, 901 Furhmann Blvd., Buffalo, N.Y. 14203 (716) 849-0735



ZX81 COMPLETE BASIC COURSE

At last, a comprehensive text for your Sinclair ZX 81! The complete BASIC Course is a manual which will immediately become an indispensible work of reference for all your ZX 81 programming.

Whether you have never done any programming or whether you are an experienced microcomputer user, the Complete BASIC Course will provide itself to you as an invaluable aid.

The Complete BASIC Course is designed to teach you to write and develop BASIC programs for the Sinclair ZX 81 – no other books or aids are necessary. All is revealed in our easy step-by-step guide with programs and "test yourself" exercises all the way through.

As you become more proficient with computing, the Complete BASIC Course will continue to be an essential guide, giving you finger tip references, numerous advanced programming techniques and memory saving devices specifically for the Sinclair ZX 81.

HOW TO WRITE PROGRAMS:

Even if the idea of writing programs is completely mystifying to you, the Complete BASIC Course will show you just how easy it is. In no time you will be able to write and enjoy complex programs for whatever use you desire.

Using the proven "TOP-DOWN" approach, the Complete BASIC Course will show you systematic and simple ways to write programs. Even experienced programmers will benefit from this Course, making programs easier to write and less prone to error!

NUMEROUS EXAMPLES:

Every concept, every function is fully described by simple programs that you can enter on your Sinclair ZX 81 in minutes.

The Complete BASIC Course contains over 100 programs and examples! These programs illustrate the use and possibilities of the Sinclair ZX 81:

- Home use
- Financial analysis and planning
- Educational applications
- Games
- Mathematical applications
- Displays of Artificial Intelligence

EVERY FUNCTION COVERED:

No matter what your application, what your confusion about any function, you will find it covered in the Complete BASIC Course.

A full and detailed discussion is included of even traditionally taboo topics such as USR, PEEK and POKE.

A handy alphabetical summary section lists all functions, and provides a short description and example programs of all topics.

A PERMANENT WORK OF REFERENCE:

The Complete BASIC Course is an excellent reference work for experienced programmers (including tips on using special techniques) as well as a comprehensive step-by-step guide for complete beginners.

The Complete BASIC Course has over 240 pages filled with information in an attractive durable ring binder – this is a lay-flat work of reference that deserves a place next to every Sinclair ZX 81 microcomputer.

$34.95

BASIC Course Programs on Cassette

All major programs in the BASIC Course are available pre-recorded in this set of cassettes. This is a valuable adjunct to the Course, saving you time and effort.
$7.50 ($9.95 in Canada)

LAST MINUTE SOFTWARE!

Backgammon (16K) Includes a high quality graphic representation of the board. Now $8.95
Flight Simulation . Incredible realistic machine code program for piloting your own ZX81! Now $9.95
Damsel & the Beast (1K) Superb adventure game. **$14.95** ($17.95 in Canada)
Two Adventure Games (16K) Now $8.95

MORE BOOKS!

Not Only 30 Programs for the Sinclair ZX81: 1K-
Not only over 30 programs, from arcade games to the final challenging Draughts playing program, which all fit into the unexpanded 1K Sinclair ZX81 but also notes on how these programs were written and special tips! Great value!
$14.95 ($16.95 in Canada)

Machine Language Programming Made Simple for the Sinclair
A complete beginner's guide to the computer's own language—Z80 machine language. Machine language programs enable you to save on memory and typically give you programs than run 10-30 times faster than BASIC programs.
$19.95 ($23.50 in Canada)

Understanding Your ZX81 ROM
A brilliant guide for more experienced programmers by Dr. Ian Logan, this book illustrates the Sinclair's own operating system and how you can use it. Includes special section on how to use machine code routines in your BASIC programs.
$19.95 ($23.95 in Canada)

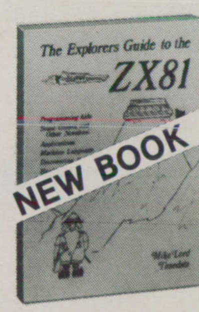
ZX81 ROM Disassembly
Part A. Covers ROM locations from 000H to OF54H and includes all functions except for routines used in floating point calculator. A must for the experienced programmer.
$14.95 ($16.95 in Canada)

64K For Sinclair ZX81 $179-95

Memotech 64K Memopak

The Memopak is a 64K RAM pack which extends the memory of the ZX81 by a further 56K. Designed to be in the price range expected by Sinclair owners. Plugs directly into the back of the ZX81 and does not inhibit the use of the printer or other add-on boards. There is no need for additional power supply or cables. The Memopack together with the ZX81 gives a full 64K, which is neither switched nor paged, and is directly addressable. The unit is user transparent, and accepts such basic commands such as 10DIM A(9000). With the Memopak extension the ZX81 is transformed into a powerful computer, suitable for business, leisure and educational use, at a fraction of the cost of comparable systems.

The Explorers' Guide to the ZX81



Programming Aids
Some Games and
Other Novelties
Applications
Machine Language
Discovering the ROM
Hardware
$9.95 ($12.95 in Canada)

GLADSTONE
Electronics 901 Fuhrmann Blvd., Buffalo, NY, 14203

Shipping and handling charge $1.50 per order books and cassettes, $3.00 per order 64k Memopak.
In Canada: **Gladstone Electronics.** 1736 Avenue Rd., Toronto, Ont. (416) 787-1448.

CHARGE TO: ☐ VISA ☐ MASTERCARD
account number: ______
Expiry date ______
Signature ______
Date ______

Please rush me:

	Quantity	Price	Total

NAME ______
ADDRESS ______
SHIPPING ______
TOTAL ______

To: Gladstone Electronics, 901 Furhmann Blvd., Buffalo, N.Y. 14203 (716) 849-0735

Call or write for free zx81 catalog

Part 3

Understanding Floating-Point Arithmetic

Ian S. Logan

In this article on floating-point arithmetic we will consider the algorithms that are used in the 8K ROM program to perform the operations of subtraction, addition, multiplication, and division. The 8K ROM program has four separate subroutines for performing these arithmetic operations. The hex addresses of their entry points in the 8K ROM (improved) are:

Subtraction: 174C
Addition: 1755
Multiplication: 17C6
Division: 1882

In each case the subroutine performs its operation on the top two floating-point numbers found on the "calculator stack," and returns the result as a single floating-point number that replaces the "lower" of the two operands. The pointer to the end of the calculator stack (STKEND) will point five locations lower after each operation in consequence.

Now, before going into the details of these four subroutines, let me recap how a floating-point representation of a decimal number is produced.

In the ZX80/81 system floating-point numbers occupy five bytes. The exponent takes one byte, and the mantissa, four.

The first stage, therefore, is to consider your decimal number in E format. It is a convention that the mantissa has only a fractional part and thereby starts with a decimal point. To illustrate the process let us take the number 456.

The decimal exponent will be +3, and the decimal mantissa will be .456. Note that the number is positive.

Next the operations should be repeated in binary to obtain a "true" binary floating point. The binary representation of 456 is 1 1100 1000. The exponent is +9 or 1001 and the mantissa will be 1110 0100 0.....

Dr. Ian S. Logan, 24, Nurses Lane, Skellingthorpe Linclon, LN6 OTT, U.K.

Now the exponent and mantissa can be made up to the correct size and expressed in hex as follows:

Exponent: 09 (1 byte)
Mantissa: E4 00 00 00 (4 bytes)

To obtain the correct Sinclair representation there are two further conventions to follow:

1) Add 128d (80h) to the exponent.

2) Replace the first bit of the mantissa—it is always set!—with a zero if the number is positive.

Hence, the final representation for 456 will be:

89 64 00 00 00 in hex, or
137, 100, 0, 0, 0 in decimal.

Now let us consider the actual subroutines.

Subtraction

The subroutine that is uniquely used for the operation of subtraction is very short as the algorithm used is to change the sign of the subtrahend (the number being subtracted) and then to proceed with an addition. The actual steps are:

1) Fetch the subtrahend and return if it has the value 0; i.e., 456 - 0 is always 456 and the operation of subtracting a zero can be ignored in all cases.

2) Fetch the first byte of the mantissa of the subtrahend and "flip" the sign bit—the first bit: 456-315 = 456 + (-315).

3) Proceed to add the two numbers.

Addition

This subroutine is fairly complicated as it has to cater for simple additions of positive numbers whose results are "within range," for additions with negative numbers, and for additions that go "out of range."

The essential parts of the subroutine are, however, fairly easy to explain. The steps are:

1) Fetch the augend (the first number of the addition). Reduce the exponent by 80h, restore bit 1 of the mantissa if the number is positive and make a note of the sign of the number. The augend is now in a "true" floating-point form.

2) Fetch the addend (the second number of the addition) and produce its "true" floating point form.

3) Compare the "true" exponents against each other and, if the addend is larger than the augend, switch over the numbers. I.e., when adding 456 + 38, the exponents in decimal are +3 and +2, so leave as is. But, when adding 38 + 456, switch over the numbers to make 456 + 38.

4) Find the "difference" between the exponents. This difference is the "amount of shift" that will be needed to "line up" the addend for the actual addition. This can be shown in decimal as follows:

456 + 38 is (.456 E + 3) + (.38 E + 2)

The difference in the exponents is +1, and the addend is shifted one place to the right to make the addition:

(.456 E + 3) + (.038 E + 3)

Now the true addition takes place between the mantissas, and the result is

(.456 + .038) E + 3 = .494 E + 3 = 494

5) Therefore shift the addend rightwards if needed and add the two numbers—the mantissas—together.

6) Normalize the result if it is not normal.

In binary floating-point the above addition of 456 + 38 becomes:

456 is 137, 100, 0, 0, 0
with a "true" form of 9, 228, 0, 0, 0
38 is 134, 24, 0, 0, 0
with a "true" form of 6, 152, 0, 0, 0

The augend is larger than the addend so the numbers do not need to be switched over, but there is a "difference" of +3, so the addend has to be shifted three places to the right to line it up with the augend. Thus

456 stays as 9, 228, 0, 0, 0
whilst 38 becomes 9, 19, 0, 0, 0

The two mantissas can now be simply added together to give the result

456 + 38 = 494 as 9, 247, 0, 0, 0

This result does not change with normalization but with the exponent augmented and the sign byte entered it becomes finally:

494 is 137, 119, 0, 0, 0

Note that, when an addition involves negative numbers, the subroutine 2's complements the negative mantissas, and that, if the final result is to be negative, it too will need to be complemented before being placed on the "calculator stack."

Multiplication

The subroutine for multiplication is fairly straightforward. For two numbers in floating-point form to be multiplied together, the exponents are simply added, and the bytes of the "true" mantissa are multiplied together bit-by-bit. The mantissas, being "fractional," when multiplied together, decrease in absolute magnitude and hence there is no danger of overflow. However, the addition of the exponents has to be checked for all the instances of "underflow"—reaching zero—and true numeric overflow.

The actual steps of the subroutine are as follows:

1) Fetch the multiplier (the first number of the multiplication) and return if it is already 0; if it is not, form the "true" mantissa.

2) Fetch the multiplicand (the second number of the multiplication) and force the result to be 0 if it is already 0. If not, again form a "true" mantissa.

3) Make available a set of registers to hold the result of the multiplication and set a counter to count the 32 times required for the multiplication of a pair of 32-bit mantissas.

4) Perform the 32 operations that produce the result.

5) Add the exponents, check for "underflow" and "overflow," and finally normalize the result.

As before, the algorithm will now be illustrated with decimal numbers before a binary example is given. Consider the multiplication of 13 * 12 to give 156. When converted to normalized E format, the problem becomes

(.13 E+2) * (.12 E+2) equals (.156 E+3)

How is it done? Following step 3 above, the result is set to be 0, and a counter set for the number of digits in the mantissas—in this case, 2. Then the following looping is performed the specified number of times.

The steps of the loop are:

1) Shift the multiplier rightwards and note the carry.

2) Increase the result by the product of the carry and the multiplicand.

3) Decrease the result by shifting it one place to the right.

Now let us look at the example values:

1st loop

1) .13 becomes .01 with a carry of 3.

2) The result, presently 0.0, is increased by 3 * .12 to become .36.

3) Decrease the result, by shifting rightward, to make .036.

2nd loop

1) .01 becomes .00 with a carry of 1.

2) The result, presently .036, is increased by 1 * .12 to become .156.

3) Decrease the result, by shifting rightwards, to make .0156.

This value of .0156 is the mantissa that goes with the exponent obtained by adding the "true" exponents, and the result is presently 0.156 E + 4 which upon normalization becomes .156 E + 3 which is the expected result.

In binary floating point 13 * 12 becomes

(132, 80, 0, 0, 0) * (132, 64, 0, 0, 0)

with the result 156 being 136, 28, 0, 0, 0

The "true" exponents of both 12 and 13 are 4, and the first nibbles (the first four bits) of the "true" mantissas are

13: the bits 1101
12: the bits 1100

In this example for simplicity the multiplication will be made between two 4-bit mantissas to produce an 8-bit result. The full operation is 32 bits by 32 bits.

Now let us consider the loops.

1st loop

1) The multiplier, 1101, is shifted to the right to give 0110 and the carry equals 1.

2) The result goes from 0000 to 1100 and is itself shifted to give 0110.

3) The end multiplier bit was set, so the present multiplicand was added to the result before it was shifted.

2nd loop

1) The multiplier goes from 0110 to 0011.

2) There is no addition of the multiplicand, but the result is still shifted to give 00110.

3rd loop

1) The multiplier goes from 0011 to 0001.

2) The result is increased by adding the multiplicand; 00110 becomes 11110.

3) This is shifted to become 011110.

4th loop

1) The multiplier goes from 0001 to 0000.

2) The result goes from 011110 to 1.00111.

3) When shifted, it becomes 10011100 (8 bits given).

The exponents are now added and the 5-byte "true" number is formed as

8, 156, 0, 0, 0

which, with the exponent augmented and the sign bit reset, is

136, 28, 0, 0, 0

and that is the floating-point representation of 156.

Division

Of all the arithmetic subroutines, division is the most complicated and the least understood. It is particularly interesting to note that the Sinclair programmer himself has made a mistake in his programming (or has copied over someone else's mistake!) for

PRINT PEEK 6352 ("unimproved" ROM, 6351)

should give 218, not 225.

To divide one number into another in floating-point form requires subtracting the exponents and dividing the mantissas bit-by-bit. Care must be taken to get the correct exponent for the result because there is a "borrow" to be taken into consideration.

The actual steps of the subroutine are as follows:

1) Fetch the divisor (the second number of the division) and give REPORT-6 if it is 0; otherwise, form the "true" floating-point form.

2) Fetch the dividend (the first number of the division) and form its "true" floating-point form. A return is made if the dividend is 0 as there is no need to divide into 0.

3) Set a counter to the number of bits in the mantissas. In the actual 8K ROM program the counter is used for 34 loops as extra accuracy is sought—but then lost because of the "programming error."

4) Perform the looping operation.

5) Subtract the exponents and adjust for the "borrow" before, finally, normalizing the result as usual.

As before, the operation will be illustrated in decimal and binary arithmetic. Consider the division of 486 by 3 to give 162 which, when converted to E format, is (.486 E + 3)/(.3 E + 1) = .162 E + 3

So, the steps are:

First, set a counter to 3 as there are three significant figures in the dividend. Second, perform the following division loop three times:

a) Subtract the divisor from the present dividend. If it does "go," count the times that it does and proceed. Else, restore the dividend and proceed with the count at 0.

b) Rotate the "times count" leftwards into the result, initially 0.

c) Shift the present dividend also leftwards and note carefully whether or not a carry is produced. If there is no carry, go to 1 on the next loop; but, if there is carry, then the divisor will definitely "go" into the dividend, and this is done directly before going to 2 for the next loop.

Now let us see the example values.

1st loop

1) .486 - .3 does go once, so make the subtraction to give a new dividend of .186 and a "times count" of 1.

2) Make the result 1.

3) Shift the dividend leftwards and note that there is carry.

2nd loop

1) The divisor is now subtracted as many times as it will go; i.e., 1.86 − .3 can be done 6 times and the dividend becomes .06.

2) The "times count" of 6 is rotated into the result which goes from 1 to 16.

3) The dividend is now shifted leftwards, with .06 going to .6 with no carry.

3rd loop

1) The present dividend is divided by the divisor, giving a "times count" of 2.

2) This "times count" is then rotated into the result which thereby goes from 16 to 162. Strictly, this should now be read as .162.

The exponents of +3 and +1 can now be subtracted from each other to give +2, the "borrow" taken into consideration to make it +3, and the final result considered as .162 E + 3.

Certain of the above mechanisms are difficult to explain in decimal arithmetic, but, by repeating the operation in binary floating-point arithmetic, I hope these points will become clear.

In binary floating-point the operation is 486 / 3 or

(137, 115, 0, 0, 0) / (130, 64, 0, 0, 0)

with the result 162 being 136, 34, 0, 0, 0. The "true" exponent for 486 is 9 and for 3 it is 2. In this case the division simplifies to .1111 0011 / .1100 0000

and there will be eight loops to consider.

1st loop

1) Trial subtract the divisor. It does go once so the dividend becomes .0011 0011.

2) The "times count" goes into the result to make it .0000 0001 and the dividend is rotated to give .0110 0110.

2nd loop

1) The subtraction does not go.

2) So the result becomes .0000 0010 and the dividend .1100 1100.

3rd loop

1) The subtraction goes, giving a dividend of .0000 1100.

2) The result becomes .0000 0101 and the dividend is shifted to give .0001 1000.

4th-6th loops

1) On no loop does the subtraction "go."

2) Hence the result becomes .0010 1000 and the dividend is .1100 0000.

7th loop

1) The subtraction goes, leaving a dividend of 0.

2) The result becomes .0101 0001.

8th loop

1) The result still has to be shifted, although the dividend has been exhausted.

2) It becomes finally .1010 0010.

The exponents are now subtracted: 9 - 2 = 7. The 'borrow" is considered to give a final exponent of 8 and this makes the result:

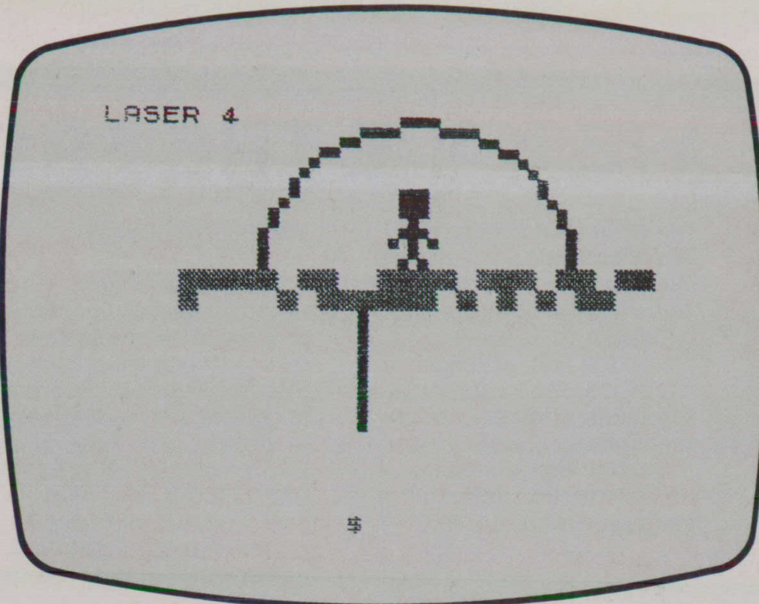
"true": 8, 162, 0, 0, 0

in Sinclair form: 136, 34, 0, 0, 0

which is decimal 162.

The reader is encouraged to try the operations for any other values. Further details of the arithmetic routines can be found in *Sinclair ZX81 ROM Disassembly, Part B* by Dr. Ian Logan and Dr. Frank O'Hara. (*Ed.—This work is available from several sources. See the ads in this issue.*)

4K ROM; 16K RAM
8K ROM; 16K RAM



ZX Destroyer

Raymond Fowkes

ZX Destroyer is a fast moving action game which is written in a hybrid Basic/machine language program to create a continuous, flicker free display. The machine language code routine is a carefully synchronized program loop which controls the laser base, alien ship, and video display. The Basic section provides the initial set-up and ending.

4K ROM Version

Loading the Machine Code

The machine code routine is placed in a REMark statement at the beginning of the program. Remember, whenever you are working with machine code, *save frequently*. It is hard to save too often!

1) Type in the lines in Figure 1 very carefully. Just enter the numbers one after another. Do not enter the spaces; these are included only for your convenience. Note also that the O's in lines 1 and 2 are not zeros. It is suggested that you SAVE after entering each REM line.

2) Type in the checker routine in Figure 2.

3) Press RUN and NEWLINE and wait for the results.

4) Unless you are one of the lucky few, the computer will indicate a mistake in one of the four REM lines. If so, go back and recheck your entry. Correct the indicated line. Repeat this procedure until the result is "PROGRAM OK." SAVE the corrected version. You are now about 90% sure that the machine code routine will run correctly.

Raymond Fowkes, PO Box 336, Coalinga, CA 93210.

5) Enter the lines in Figure 3 over the existing ones and RUN. The machine code program has now been processed and stored in line 1. *Do* **not** *LIST line 1 or press HOME!* Doing so could crash the program and ruin all the work you have just done. SAVE.

Entering the Basic Program

1) Erase lines 2-120 by entering the line number and hitting NEWLINE.

2) Enter the program in Figure 4 exactly as printed. This provides the set-up for the game. SAVE again.

3) Press RUN and NEWLINE.

4) In the center of your display you should see the alien space ship with the alien in the middle of his control dome. Two powerful rotating screens protect the bottom of the ship. You will also see the invader's laser moving back and forth underneath the ship. It will fire when your laser base comes into range.

5) In the lower left corner of the display you will see a "$" which represents one of your laser bases. In the upper left corner the display will tell you how many functioning lasers you have.

6) If the screen display does not match the description in 4) and 5) above, go back and check your Basic program or reenter your machine code routine. If neither of these works, wait for the next Glitchoidz report!

8K ROM Version

The 8K ROM version is very similar to the 4K ROM version and gives flicker free action in the FAST mode.

Loading the Machine Code

1) Load the the machine code in Figure 6 as in step 1 in "Loading the Machine Code" above. This code is a modification of the 4K ROM version. Unlike the 4K machine the 8K machine allows you to LIST the REM lines without crashing the system.

2) Enter the checker program in Figure 7. RUN the program in FAST mode.

3) After several seconds of processing, the computer will indicate a "PROGRAM OK" or a mistake in one of the first four lines. If you have 2K RAM, editing requires a special process. First, LIST the line with the mistake. Then execute a CLEAR command and EDIT when the screen is cleared. You can then make the correction and return the line to the program by pressing NEWLINE. RUN again and make any further corrections needed until the result is "PROGRAM OK."

4) Enter the lines in Figure 8 over the existing lines. SAVE.

5) RUN the program. When it is finished, the machine code routine will be processed and stored in line 1.

6) Delete lines 20-90 by entering the line numbers and NEWLINE. Then enter the lines in Figure 9. Line 10 causes CLS to fill the screen with spaces. If you have 16K RAM, this is automatic, and you can use this line as a place for the program name. Lines 100-170 set up the display. Lines 180-200 reset the laser indicators and start the laser base at the lower left corner of the display. Lines 210 and 240 should be entered only if your computer can run in the SLOW mode.

7) Again SAVE. Then RUN. You should see the same screen display as above in the 4K version.

Figure 1: 4K ROM Machine Code.

```
1 REM G41 923 111 117 CY9 U2A S47 Q8D UT8 FXV
22S N59 FWV 22S NAA WYS MAA 9CM 3SX F71 FTC J1X
4CM 2QC F1C 94C P57 28D UAK 1T9 ETK QHR CF4 K3Q
T1T 1QS DQG 75X BCM 5SY CF5 729 A9C M4T SX3 CDT
CX3 5DO YCG 6XT CK3 242 C64 IY1 QS6 DUA ZFU K37
UPV IY1 8FU QCI 3CT 3MX XSE 7EG K3X S8C K63 Y81
S6D UAJ STG QY3 3AK E87 5HA SFC G37 LXG CK3 08

2 REM FUI CAQ 7WK 1ZF UPI K1S 271 CG3 XV7 1CK
31H A1C GB6 48H UK1 IEB QS3 7TG 67Z HUC IBS 97N
G6Z MPS H7L CGB 6IS E7K CGB YNK 77I PHK 1CK B1P
97B VAK 1QH 7R9 T27 ZXQ CKB 2C6 643 IEB QS6 DUA
ZHU K77 P9H ACJ C7Q X2C KC7 8DU IC1 QMF C12 C1S
2JI X1Q MPO 9FO SAJ 7BH ACP 6N3 8DU QK1 I11 9TY
X2S TIX 2KQ 187 K67 SD7 IV1 KEH ACM 6RC F6L PU

3 REM 211 116 111 111 712 278 231 121 268 116
612 712 116 612 478 162 778 174 151 412 127 211
474 127 417 317 311 121 212 162 361 144 741 211
626 212 426 111 174 172 741 111 317 416 311 616
311 716 576 115 721 121 741 111 173 121 122 211
113 111 222 172 176 141 221 116 211 177 216 111
731 212 216 616 112 171 118 217 611 111 721 42

4 REM 221 481 112 112 211 212 512 761 171 272
111 867 612 322 221 178 121 116 221 227 212 111
621 212 111 761 212 111 761 222 111 122 172 111
213 182 111 861 631 111 721 174 121 178 173 121
122 211 111 872 113 127 211 212 161 143 321 117
611 111 433 211 761 111 873 111 212 121 111 163
111 631 111 661 121 731 111 211 874 147 416 22
```

Figure 2: 4K ROM Checker Program.

```
  10 LET B=0
  20 FOR A=16426 TO 17419
  30 LET B=B+PEEK(A)-PEEK(A+1)
  40 IF A=16672 THEN IF NOT B=36
5 THEN GOTO 120
  50 IF A=16920 THEN IF NOT B=65
THEN GOTO 120
  60 IF A=17170 THEN IF NOT B=35
9 THEN GOTO 120
  70 LET A=A+1
  80 NEXT A
  90 IF NOT B=22 THEN GOTO 120
 100 PRINT "PROGRAM OK"
 110 STOP
 120 PRINT "MISTAKE IN LINE#";A/
249-65
```

Figure 3.

```
  10 LET B=16427
  20 FOR A=16427 TO 16916
  30 POKE A,PEEK(B)+PEEK(B+498)*
35-1044
  40 LET B=B+1
  50 IF B=16672 THEN LET B=B+4
  60 NEXT A
  70 POKE A+3,254
  80 POKE A+2,2
  90 POKE A+1,0
 100 POKE A,118
 110 POKE 16403,2
 120 LIST 2
```

Figure 4: 4K ROM Basic Program.

```
 100 POKE 16421,24
 110 PRINT "LASER#5#########WW"
 120 LET B$=CHR$(131)
 130 PRINT ,"####W";B$;B$;"##";B
$;B$;"W"
 140 PRINT ,"##E";B$,"####";B$;"
D"
 150 PRINT ,"#FE",""#####RD"
 160 PRINT ,"#S######";CHR$(128);
"Q#####";CHR$(136)
 170 LET B$=CHR$(130)
 180 PRINT ,B$;"######";CHR$(134
);"E######Q"
 190 PRINT ,"Q#####F";CHR$(134);
CHR$(136);"######";B$
 200 PRINT ,"Q######SD######";B$
,"####AAAAAAAAAAAAAAAAAAAAAAAA",
"####AAAAAAAAAAAAAAAAAAAAAAAA",
 210 FOR A=1 TO 14
 220 PRINT ,,,,
 230 NEXT A
 240 PRINT
 250 POKE 16429,0
 260 POKE 16432,177
 270 POKE 16437,0
 280 RANDOMISE USR(16439)
 300 INPUT A$
 310 CLS
 320 RUN
```

Figure 5: Sample Opening and Ending.

```
  10 CLS
  20 PRINT ,"ZX"
  30 PRINT
  40 PRINT
  50 PRINT "SKILL:##30=BEGINNER#
TO 1=EXPERT"
  60 INPUT A
  70 IF 1>A OR A>30 THEN GOTO 60
  80 POKE 16428,A
  90 CLS
 300 IF PEEK(16437)=16 THEN GOTO
340
 310 CLS
 320 PRINT "YOU WERE DESTROYED"
 330 GOTO 400
 340 INPUT B$
 350 CLS
 360 PRINT "*#INVADER DESTROYED#
*"
 370 PRINT
 380 PRINT "CRASH CURSOR"
 390 PRINT "WOULD BE PROUD OF YO
U"
 400 PRINT
 410 PRINT
 420 PRINT "N/L TO PLAY AGAIN"
 430 INPUT B$
 440 IF B$="" THEN RUN
     (or THEN RUN 90)
```

Note: To make winning more difficult, add this and delete line 70 (to prevent memory overflow):

```
210 FOR A=1 TO 52
220 PRINT "++++++++";
235 PRINT ,,,.
```

Figure 6: 8K ROM Machine Code.

```
   1 REM G41A2E111117CYQU2B8DUT8
2W257V739230SMV7392W25MV735NAAWY
5MAA9CM35XF71PTCJ1X4CM20CF1C94CF
5728DUAK1T9ETKQHRCF4K6QT1T1Q5DQG
75XBCM55YCF5729A9CM4TSX30DTCX35D
O9CG6XTCK3242C64IY1Q56DUAZWUK67U
PPIY18WUQCI3CT3MXRSE7EGK6XMKCK6E
Y815GDUAJSTGGQ74HASFCG37LXGCK30
   2 REM 8WUICAQ7WK1ZWUPIK1S271C
G3XV71CK31HA1CGB648YUK1IEBQ567TG
67ZYUCIBS77NG6ZMPSH7LCGB6ISE7KCG
BMNK27IPHK1CKB1P97AV9K1QH7R9T27Z
XQCKB2C664BIEBQ56HUAZYUK27P9HACJ
C7QX2CKC78DUIE2QMFO1201S2JIX1QMF
O9FOSAJ7BHACP6NB8HUQK1I119TYX2ST
IX2KQ187K67SD7IV1KEHACM6RCF6L7U
  20 REM 21131711111171427821268
11128661211127661211127661247816
27781741514121272114741274173173
11121212162361144741211626212426
11117417274111131741631161631171
65761157211217411111731211422111
13111242172176141221116211177217
1117312122167611182176111117214
  30 REM 24214811121142112125127
51171272111867612324221178121116
22142721211162121211176121211176
12221111221721112131821118616311
11721174128178173121142211111872
11312721121211114332111761111143
321117611118731182121211111631116
3111166112173111121187414741652
```

Figure 7: 8K ROM Checker Program.

```
  50 LET B=0
  60 FOR A=16509 TO 17516 STEP 2
  70 LET B=B+PEEK A*(PEEK (A+1)-
1)
  80 LET C=(A-16507)/126
  90 IF C=2 AND B<>243529 OR C=4
 AND B<>488940 OR C=6 AND B<>612
608 THEN GOTO 140
 100 NEXT A
 110 IF B<>733561 THEN GOTO 140
 120 PRINT "PROGRAM OK"
 130 STOP
 140 PRINT "MISTAKE IN LINE ";"1
2 2030"(C-1 TO C)
```

Figure 8.

```
  10 REM
  20 REM
  30 REM
  35 REM
  40 POKE 16511,244
  50 POKE 16512,1
  60 LET B=16514
  70 FOR A=B TO 17005
  80 IF B=16760 THEN LET B=B+6
  90 POKE A,PEEK B+PEEK (B+504)*
35-1044
 100 LET B=B+1
 110 NEXT A
 120 POKE 16511,238
 130 POKE A+5,234
 140 POKE A+4,0
 150 POKE A+3,2
 160 POKE A+2,10
 170 POKE A+1,0
 180 POKE A,118
```

Figure 9: 8K ROM Basic Program.

```
 10 POKE 16389,128
 90 CLS
100 PRINT "LASER 5";TAB 15;"▄▄"
110 PRINT TAB 12;"▄▄  ▄▄"
120 PRINT TAB 10;"▀";TAB 20;"▀"
130 PRINT TAB 9;"▞";TAB 21;"▚"
140 PRINT TAB 9;"▞     █     ▚"
150 PRINT TAB 8;"▌     ▀▀     ▐"
160 PRINT TAB 8;"▌     ▟▙     ▐"
170 PRINT TAB 8;"▌     ▟▙     ▐";TAB 4;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";TAB 4;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
180 POKE 16516,0
190 POKE 16519,223
200 POKE 16524,0
210 FAST
220 RAND USR 16526
230 PAUSE 99
240 SLOW
250 RUN
```

The number or letter indicates the key on which the graphic is found and the number in () gives the number of times to press the key.

```
100  (2) 6
110  6; (2) 7; (2) 7; 6
120  3; 7; 7; 4
130  3; 1; 2; 4
140  T; space; 5; Y
150  8; R; 1; 5
160  5; 3; R; Y; 8
170  5; T; 4; 8; (24) A; (24) A
```

Figure 10: Changes for Game Variations.

To change:	4K ROM	8K ROM
Number of laser bases:		
Change the number in line	110	100
Speed of alien's laser:		
POKE a number from 1-30 into	16428	16515
Speed of alien's screens:		
POKE another number into	16830	16919
(It is 6 now; try 1 or 2)		
Laser base graphic:		
POKE new character code into	16621	16716
and into	16904	16993
Alien's laser graphic:		
POKE new character code into	16527	16622

Figure 11: Variables.

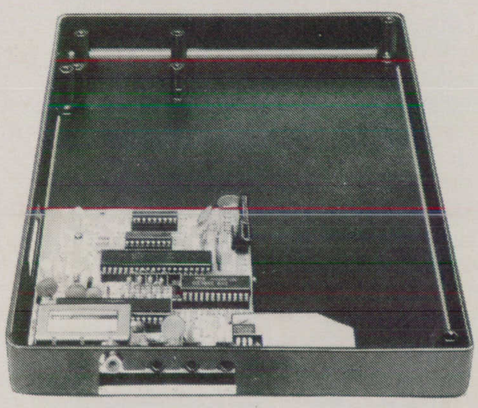
Decimal Address	d: (IX+d)	Initial Setting	Function
16427	0	50	skill variable
16428	1	3	skill set
16429	2	0	laser indicator
16430	3	8	gun position
16431	4	1	gun direction (1,-1)
16432	5	177	base position
16433	6	0	laser position
16434	7	0	laser position
16435	8	0	"shell" position
16436	9	0	"shell" position
16437	10	0	"shell" indicator
16438	11	6	rotate counter

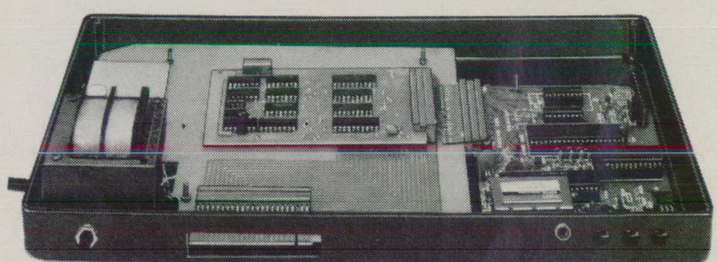
FULLER FD SYSTEM $79.95

Professional Keyboard & Case for Sinclair ZX81

The ZX8l fits inside.
The tough ABS injection moulded plastic case measures 8″ x 14″ x 2½″ and hooks up to your ZX printed circuit board in minutes. No technical know how or soldering is required.

KEYBOARD LAYOUT:
All the Sinclair ZX81 keys are duplicated on our layout, with extra shift and new line keys. The professional momentary action key switches have a guaranteed life of 10^6 operations. The unit is fully built tested and comes complete with a money back guarantee.





INSTALLATION
Simply unscrew the ZX printed circuit board from its case and screw it into the FD Case.

MOTHERBOARD:
We also manufacture a mother board which allows expansion to the ZX memory and I/0 facilities WITHIN the case, as well as our power supply unit and reset switch.

AD Code	

	Price †	Qty	Amount
Fuller FD System 42 Keyboard & case	79.95		
FD System Motherboard	29.95		
FD 16K. Memory Module	49.95		
FD 64K. Memory Module	149.96		
FD PSU 9 Volts at 2 amp.	14.95		
FD *Shipping and Handling*	6.00		

Mail to **FULLER MICRO SYSTEMS,**
The ZX Centre, Sweeting Street, Liverpool 2. England, U.K.

Name ....

Address ....

City/State/Zip ....

† US Dollars

Variations

Variations can be made in both versions. Figure 10 summarizes some of the main possibilities. If you have extra RAM, you can add your own openings and endings such as suggested in Figure 5. Note that PEEK 16524=16 can indicate to a program that the invader was destroyed.

British TV Differences

Since the speed of British television receivers is different from that of U.S. receivers, some adjustment may be needed. If this adjustment cannot be made with the vertical hold, then some changes in the program need to be made to slow down. For the 8K ROM try POKE 16538,79, and POKE 16556,79; for the 4K ROM, POKE 16455,80, and POKE 16464,79. These will slow it down quite a bit, but other values may work better. (*Ed.*—**SYNC** *would be interested in hearing from British readers concerning their experience with this problem.*)

Playing the Game

The goal of the game is to blast through the bottom protective screens of the alien ship with your lasers and hit the alien (the left foot). This will destroy the alien and his ship. The game begins when you launch your attack on the alien ship from your laser base. Since both your laser and the alien's laser fire only on the perpendicular, you must get your base in position to hit the ship. To move your base, press the arrow keys: 5 for left, and 8 for right. To fire press 0 or 1. Your laser has an unlimited number of shots. The alien's laser has computer controlled accuracy and will destroy all five of your laser bases if you are not careful. So watch out!

The game is over when you have destroyed the alien or when you have lost all your laser bases. When the game is over, you can play again by pressing NEWLINE. If not, press NEWLINE and BREAK and you will return to Basic. You can also BREAK at any time during the game.

Figure 12: Z80 Instructions (4K ROM)

```
Label   Instruction       Comment
START   LD IX, 16427      index register points to variables
DISP    OUT (254),A       display one frame
        LD A,73
        LD B,25
        LD HL,(16396)
        SET 7,H
        LD C,32
        CALL 432
        LD A, 232
        INC B
        DEC HL
        LD C,31
        CALL 432
        LD A,127          set mask for BREAK key
        IN A,(254)        read keyboard
        RRA               push break bit into carry
```

Label	Instruction	Comment
	RET NC	return to BASIC if BREAK pressed
	LD A,231	set mask for top row
	IN A,(254)	read keyboard
	EX AF,AF'	save in alternate register
	LD A,(IX+2)	test laser indicator
	AND A	
	JR NZ,FIRE	continue laser fire
	LD B,35	set sync loop
	JR CPGUN	
MOVE	DEC (IX+0)	decrement skill counter
	JR NZ,LEAP	move laser gun if zero
	LD A,(IX+1)	restore counter
	INC A	
	LD (IX+0),A	
	LD C,(IX+3)	BC holds relative gun position
	LD E,(IX+4)	E holds direction (-1 or 1)
	LD B,1	
	LD HL,(16396)	HL points to display file
	ADD HL,BC	HL holds gun position
	LD (HL),0	erase gun
	SRA E	push off bit 0
	LD D,E	get D same sign as E
	RL E	restore E
	ADD HL,DE	HL holds new position
	LD A,C	put old pointer into A
	ADD A,E	update A
	LD (IX+3),A	replace pointer
	LD (HL),2	put gun in new position
	ADD HL,DE	move pointer 5 spaces
	SLA E	
	SLA E	
	ADD HL,DE	
	LD A,117	load A with "newline"
	INC A	
	CP (HL)	compare with new location
	LD B,4	sync
	JR NZ,CPGUN	reverse direction if necessary
	LD A,(IX+4)	get old direction
	NEG	negate it
	LD (IX+4),A	replace it
	LD B,1	sync
CPGUN	EX AF,AF'	get keyboard reading
	LD C,A	put into C
	EX AF,AF'	save it
	LD A,(IX+3)	get gun position
	BIT 4,C	test "left" key
	JR NZ,LESS	add 12 if "left" pressed
	ADD A,12	
LESS	BIT 2,C	test "right" key
	JR NZ,MORE	subtract 12 if "right" pressed
	SUB 12	
MORE	ADD A,173	add offset
	CP (IX+5)	compare with base position
	JR NZ,MOVE	begin laser if gun in position
	LD (IX+2),1	set laser indicator
	LD H,1	HL holds gun position
	LD L,(IX+3)	
	LD DE,33	move down one line
	ADD HL,DE	
	LD BC,(16396)	BC points to display
	ADD HL,BC	adjust HL
	LD (16433),HL	store in memory
	LD (HL),2	fill with laser character
	LD B,29	sync
LEAP	JR SHOOT	
FIRE	LD DE,33	
	LD HL,(16433)	get old laser position
	ADD HL,DE	move down one line
	INC (IX+2)	increment indicator
	BIT 7,(IX+2)	test bit 7 of indicator
	JR NZ,ENDF	erase laser if set
	LD A,13	check for laser hit base
	LD B,13	sync
	CP (HL)	
	LD (HL),2	fill with laser character

ZX81 or TIMEX 1000

data-assette®

Your One Stop Shop
All Products Guaranteed

EVERYTHING YOU WILL NEED!! ADD ON TO YOUR ZX 81/TIMEX 1000!! ALL PRODUCTS 100% GUARANTEED !! NOW GROW YOUR SYSTEM AS YOUR NEEDS CHANGE, AND WE WILL GIVE YOU PRODUCTS, SERVICE, SOFTWARE AND ADVICE. WE HAVE BEEN IN THE BUSINESS IN ENGLAND FOR OVER ONE YEAR AND OFFER THIS EXPERIENCE TO ALL OUR CUSTOMERS.

New Low Prices

1st STEP - MEMORY - RAM

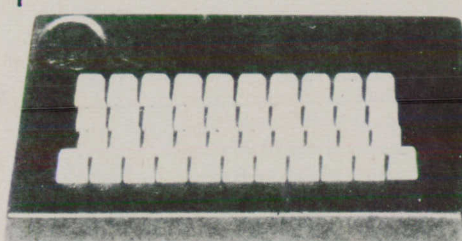
16K - 32K or 64K

$59.95 - $109.95 or $149.95

Each RAM extends the BASIC memory to provide the user space for all applications.

2nd STEP - KEYBOARDS

FULL SIZE or MINI $75.00



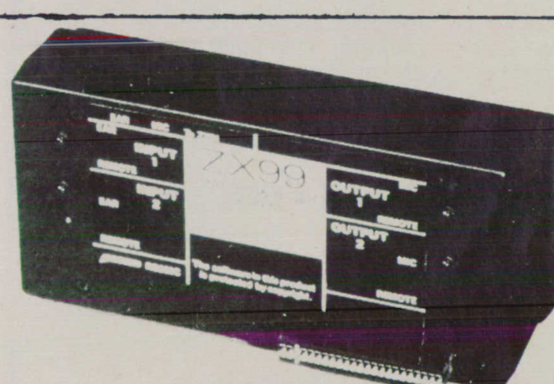
Real Professional KEYBOARDS Computer Size and with special features like!!!!Repeat Keys on the full size. Both units are easy to install at home . Supplied with cables & instructions.

3rd STEP - ZX-99 TAPE CONTROL

Now add Real "DATA-PROCESSING" power to your ZX-81 with our sophisticated extension. Provides "FULL SOFTWARE CONTROL" of four (4) tape decks and a PRINTER interface for any RS 232 Serial Printer, giving full 132 ASC II characters of output. Plus!!!!!!

★ ★ AUTOMATIC "TAPE TO TAPE COPY"

★ ★ TAPE BLOCK SKIP, DIAGNOSTIC ASSISTANCE



$150.00

INTRODUCTORY OFFER "$375.00"

Special to New Owners, show us your invoice and order in 30 days from the date. We will give you this "FULL PACKAGE" for the low price of $300.00. 64K RAM + Keyboard + ZX99. Add it to your printer and for under $400.00 you have full MINI-CAPABILITIES.

★ ★ DATA RETREIVAL WORD PROCESSING

★ ★ RS 232 C Serial Printer Interface

EQUALS REAL MINI-COMPUTER CAPABILITIES

BLANK CASSETTES

C-5	$.69
C-10	$.74
C-12	$.77
C-15	$.79
C-20	$.84
C-25	$.89

plus Shipping

data-assette®
56 South 3rd Street
Oxford, PA 19363

Telephone Orders
1-800-523-2909
In Pennsylvania
(215) 932-3626

CHARGE TO:
VISA MASTERCARD
account number
Expiry date
Signature
Date
NAME
ADDRESS

Please rush me:

Quantity	Price	Total

SHIPPING
TOTAL

SYC-1

Send me details:

- ☐ Software
- ☐ Supplies
- ☐ User Clubs
- ☐ Local Dealers
- ☐ New Products

Hi-Resolution or Color Boards

Figure 12: Z80 Instructions (continued)

```
Label      Instruction        Comment
           JR NZ,MISSED
           EXX                save the registers
           LD (IX+5),177      replace laser base at left
           LD HL,7
           LD BC,(16396)
           ADD HL,BC          HL points to (D-FILE)+7
           DEC (HL)           one less laser base
           LD A,28            check for "zero"
           CP (HL)
           RET Z              return to BASIC if zero

           LD HL,689          point to bottom left of screen
           ADD HL,BC
           LD (HL),13         fill with base character
           EXX                restore registers
           LD B,4             sync
MISSED     DJNZ -2            timing loop
           LD A,14            check for laser at bottom
           CP (IX+2)
           LD B,20            sync
           JR NZ,REPLACE
           LD (IX+2),128      indicate laser ending
           LD HL,(16433)      get old laser pointer
           LD DE,-396         move up 13 lines
           ADD HL,DE
           LD B,31            sync
           LD (HL),0          erase top of "laser beam"
REPLACE    LD (16433),HL      update pointer
           JR SHOOT

ENDF       LD (HL),0          continue erasing laser
           LD A,(141)         check for laser fully erased
           LD B,35            sync
           CP (IX+2)
           JR NZ,REPLACE
           LD B,35            sync
           LD (IX+2),0

SHOOT      DJNZ -2            timing loop
           XOR A              zero A
           CP (IX+10)         compare shell indicator
           JR Z,READ

           LD HL,(16435)      HL holds shell pointer
           LD (HL),0          erase shell
           LD DE,-33          move up one line
           ADD HL,DE
           LD A,2             check for laser hit shell
           LD B,28            sync
           CP (HL)
           JR Z,RESET
           LD (16435),HL      update pointer
           INC (IX+10)        increment counter
           LD A,8             check for shell hit alien
           LD B,22            sync
           CP (HL)
           JR Z,RETN          if shell hit alien
           JR C,BITE          if shell hit ship
           LD A,16            check for shell at top
           LD B,20            sync
           CP (IX+10)
           JR Z,RESET
           LD A,13            don't show shell if 3 below top
           LD B,19            (so shell isn't carried away
           CP (IX+10)          by rotation at bottom of ship)
           JR C,ROTATE
           LD (HL),6          fill new shell location
           LD B,17
           JR ROTATE

BITE       LD (HL),0          take chunk out of ship
RESET      LD (IX+10),0       reset indicator
           JR ROTATE

RETN       LD B,80            loop 80 times
           LD E,-2            DE=-2
AGAIN      LD (HL),0          erase a character
```

Label	Instruction	Comment
	ADD HL,DE	move left 2 spaces
	DJNZ AGAIN	do again for exploding effect
	RET	return to BASIC
READ	EX AF,AF'	get keyboard reading
	BIT 0,A	test "fire" key
	LD B,34	sync
	JR NZ,SAVE	
	LD (IX+10),1	set shell indicator
	LD L,(IX+5)	HL holds relative base position
	LD H,2	
	LD DE,-33	move up one line
	ADD HL,DE	
	LD BC,(16396)	BC points to display file
	ADD HL,BC	adjust HL
	LD (16435),HL	update pointer
	LD (HL),6	fill with shell character
	LD B,24	sync
SAVE	EX AF,AF'	replace keyboard reading
ROTATE	DJNZ -2	syncronization loop
	DEC (IX+11)	decrement rotate counter
	LD B,95	sync
	JR NZ,SLIDE	
	LD (IX+11),6	restore rotate counter
	LD HL,(16396)	point to display file
	LD DE,198	
	ADD HL,DE	HL points to top left of bottom of ship
	LD A,(HL)	save first character
	LD D,H	
	LD E,L	DE=HL
	INC HL	HL=DE+1
	LD BC,23	shift 23 spaces
	LDIR	
	LD (DE),A	replace first character at right
	LD DE,32	move to bottom right
	ADD HL,DE	
	LD A,(HL)	save last character
	LD D,H	
	LD E,L	DE=HL
	DEC HL	HL=DE-1
	LD C,23	shift 23 spaces
	LDDR	
	LD (DE),A	replace last character at left
	LD B,10	sync
SLIDE	DJNZ -2	timing loop
	LD E,(IX+5)	
	LD D,2	
	LD HL,(16396)	
	ADD HL,DE	HL holds base position
	LD (HL),0	erase base
	LD DE,0	zero DE
	EX AF,AF'	get keyboard reading
	BIT 4,A	test "left" key
	JR NZ,RIGHT	
	DEC DE	DE=-1 if "left" pressed
RIGHT	BIT 2,A	test "right" key
	JR NZ,LEFT	
	INC DE	DE=0 or 1 if "right" pressed
LEFT	ADD HL,DE	move right or left
	XOR A	zero A
	OR (HL)	test for space and reset carry
	LD B,19	sync
	JR Z,MERGE	
	SBC HL,DE	move back
	LD B,17	sync
	LD E,0	zero addend
MERGE	LD (HL),13	replace laser base
	DJNZ -2	timing loop
	LD A,(IX+5)	get old pointer
	ADD A,E	update,
	LD (IX+5),A	and replace
	JP DISP	jump back to beginning

4K ROM; 1K RAM

Galaxy Invaders

Phil Gervais

In *Galaxy Invaders* you are under attack by a fleet of enemy spacecraft which you must destroy before they destroy you. This game uses the ZX80's 1K RAM, graphics, and screen blackout characteristics to produce an arcade type game. The program (see Listing 1) is fairly simple, but it employs techniques which are used in the popular arcade games: 1) insufficient information (you know they are coming, but you do not know when); 2) the ability to play all day (if you are good enough); and 3) a running point total (for self-satisfaction).

A typical game goes like this: First, the title block appears. You climb aboard one of the three attack ships in your squadron, prepare yourself for the worst, hit NEWLINE, and blast off into space. Suddenly, you see the enemy ships approaching! You immediately hit your firing button (NEWLINE) as quickly as possible. The enemy craft disappear quickly, but not before you hit one of their small reconnaissance ships with a deadly phaser. Your onboard computer evaluates your shot. Each ship you hit is worth either one or two points. Since your objective is to destroy as many ships as possible, a high point total is the measure of a successful mission. Your new point total is displayed along with the number of ships you have hit. NEWLINE returns you to outer space for the next round of the battle.

We must note some of the features of the equipment that shape the battle strategy. Both squadrons are equipped with advanced radar firing which insures pinpoint accuracy. However, the enemy's phasers are capable of jamming your detection gear and vice versa. This insures your accuracy and disrupts the enemy's firing.

The scores in this game generally range from 50 to 100. If you get over 150, you rank among the best space pilots in the galaxy. However, if you do not seem able to win a battle, you will have to have your engineers redesign the capability of your attack ships by increasing the value of C in lines 80-90 by 20 or more.

For those who have more than 1K RAM the step in Listing 2 gives a bonus ship at 250 points.

May the luck of the galaxy warriors be with you! ■

Phil M. Gervais, 714 5th Ave. South, Clinton, IA 52732.

Listing 1: Galaxy Invaders; 4K ROM, 1K RAM.

```
  2 RANDOMISE
  4 CLS
  6 LET P=0
  8 LET Z=3
 10 PRINT ,"GALAXY INVADERS"
 12 GO SUB 76
 14 CLS
 16 FOR I=1 TO 20*RND(100)
 18 NEXT I
 20 POKE 16414,0
 22 POKE 16415,0
 24 GO SUB 70
 26 LET A=PEEK(16414)
 28 LET B=PEEK(16415)
 30 LET C=256*B+A-4
 32 CLS
 34 IF C<8 THEN GO TO 62
 36 IF C<9 THEN GO TO 94
 38 IF C<13 THEN GO TO 97
 40 PRINT "YOU WERE HIT BY PHAS
ER"
 42 LET Z=Z-1
 44 IF NOT Z=1 THEN PRINT Z;"#S
HIPS LEFT"
 46 IF Z=1 THEN PRINT Z;"#SHIP#
LEFT"
 48 IF NOT Z=0 THEN GO TO 101
 50 PRINT "POINTS ACCUMULATED:#
";P
 52 PRINT "START OVER IN NEW GA
LAXY?"
 54 INPUT W$
 56 IF CODE(W$)=62 THEN RUN
 58 CLS
 60 STOP
 62 PRINT "DESTROYED ENTIRE FLE
ET"
 64 PRINT "GOOD WORK."
 66 LET P=P+4+RND(3)
 68 GO TO 99
 70 CLS
 72 PRINT ,"SHOOT NOW"
 74 PRINT ,"=====#==="
 76 PRINT
 78 PRINT
 80 PRINT,,"SD"
 82 PRINT
 84 PRINT,"##A"
 86 PRINT,"TAAAT"
 88 PRINT, "#G#G",,"SD"
 90 PRINT
 91 PRINT "SD"
 92 INPUT W$
 93 RETURN
 94 PRINT "DESTROYED SMALL GROU
P"
 95 LET P=P+2+RND(2)
 96 GO TO 99
 97 PRINT "DESTROYED 1 SMALL CR
AFT"
 98 LET P=P+RND(2)
 99 PRINT "POINTS:#",P
101 INPUT W$
102 GO TO 14
```

Listing 2: Bonus Ship; 4K ROM, over 1K RAM.

```
  3 LET X=0
100 IF P>249 AND X=0 THEN GO S
UB 104
104 PRINT "** BONUS SHIP **"
106 LET Z=Z+1
108 LET X=1
110 RETURN di %
```

Micro Invaders *Cyril B. Smith*

In *Micro Invaders*, a space fantasy game in Basic for the 8K ROM, 1K RAM Sinclair, the Earth is under attack by a fleet of Micro Invaders from outer space.

You are in charge of a mobile intercepter launcher defending the Earth. Intelligence has reported that the invading ships can be destroyed only by hitting them just inside the left landing leg. This requires pinpoint aiming. You control your launcher movement with the arrow keys using 5 for left and 8 for right. You launch your intercepters by pressing the zero key.

Cyril B. Smith, 4737 Shadowglen Dr., Colorado Springs, CO 80907.

```
 10 LET G=0
 15 LET G=G+5
 20 LET B=15
 25 LET N=5
 30 LET L=0
 35 DIM A(5)
 40 GOTO 60
 45 CLS
 50 LET N=N-1
 55 IF N=0 THEN GOTO 15
 60 LET A=20
 65 LET C=0
 70 IF L=20 THEN GOTO 200
 75 FOR M=1 TO N
 80 LET A(M)=INT (RND*18+7) .
 85 NEXT M
 90 FOR I=1 TO 12
 95 FOR M=1 TO N
100 PRINT AT L,A(M);"▗▀▖"
105 IF A=L AND B=A(M) THEN GOTO
45
110 NEXT M
115 IF INKEY$="0" THEN LET C=1
120 IF C=1 THEN GOTO 140
125 IF INKEY$="5" THEN LET B=B-
1
130 IF INKEY$="8" THEN LET B=B+
1
135 GOTO 145
140 LET A=A-2
145 PRINT AT A,B;"▄█▄"
150 IF A<L THEN LET C=0
155 IF A<L THEN LET A=20
160 PAUSE 25
165 CLS
170 NEXT I
175 LET L=L+4
180 GOTO 70
200 PRINT AT 21,20;G-N;" INVADE
RS"
```

The invaders come swooping in from outer space in formations of five space ships. The ships can change formation at times, and they can hide behind each other. Hitting one ship results in a formation rearrangement and a delay in renewing the attack.

If all the ships of the formation are destroyed before they reach you, you are assigned a new formation to deal with. The number of craft destroyed is shown in the lower right of the display. The game is over if your launcher is hit by the invaders.

Since the program is quite full, more memory would be needed to add anything unless the efficiency of the program can be improved.

Program notes:
Line 100: graphics on T and Y.
Line 145: graphics on Q and 4.

8K ROM; 1K RAM

Comet Crusher

Chuck Dawson

FLASH!!! Astronomers have spotted a hitherto unknown comet a million kilometers from the earth. Calculations indicate that it will collide with the Earth's atmosphere at a shallow angle and will probably break up into many pieces. Each of these pieces could destroy a city. Top scientists have advised placing special Comet Crusher missiles at strategic locations around the world with the hope of blasting the comet fragments into dust before they can hit any of the populated areas.

Leaders of countries around the world have issued an urgent call for volunteers to man the Comet Crusher missile batteries. Those who apply must have a keen sense of timing and iron nerves, because millions of people will be depending on them. Physical strength is not required, for one need only push the launch button to place the missile close to the fragment as it streaks overhead.

As a successful applicant you have been assigned to one of the launch sites with the following instructions:

1) Check your ZX81 computer. If it has over 1K RAM, enter the program in Listing 1. Note the following lines:

30 graphics on 3, 8, T, 4.
50 graphics on S
110 leave 17 spaces inside the " ".
140 graphics on 3, 8, T, 4, and asterisk.

2) If your computer has 1K RAM, enter the program in Listing 2. Note the following line:

2 graphics on D.

3) If your computer is a ZX80 with 8K ROM, add these lines to Listing 1:

85 PAUSE 45
86 POKE 16437,255

4) After entering your program, SAVE it *before* you run it.

5) Hit SLOW and ENTER; then hit RUN and ENTER.

6) After your viewing screen is functioning, you are ready to launch your pulverizing missiles against the assigned fragments. To fire press F.

7) Hits are recorded and reported to the coordinating computer to make sure that no fragments will slip through the world-wide network. ■

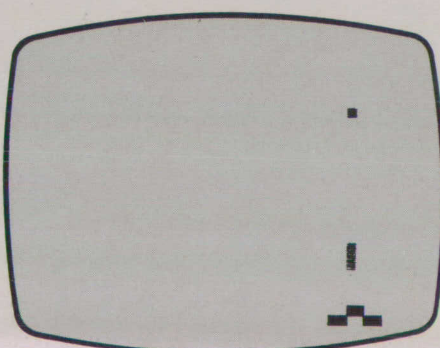
Chuck Dawson, 6520 Victoria, Fort Worth, TX 76118.

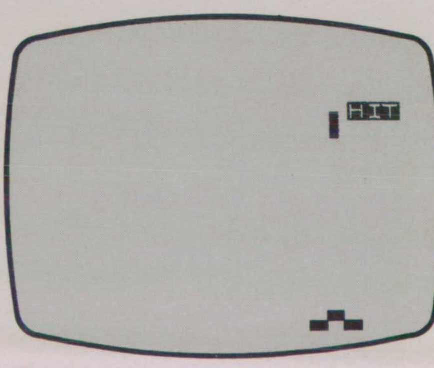
Listing 1: Comet Crusher (8K ROM; over 1K RAM).

```
   1 REM "COMET CRUSHER" TO SAVE
 GOTO 500
   5 CLS
  10 LET H=18
  15 LET A=0
  20 LET Y=37
  25 LET X=0
  30 PRINT AT 18,19;"▗";TAB 19;"
▌";TAB 19;"▟▖"
  40 PRINT AT 20,0;"PRESS F TO F
IRE"
  50 PRINT AT 21,0;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
  60 UNPLOT X-1,Y
  70 PLOT X,Y
  80 LET X=X+1
  90 IF X>63 THEN GOTO 1
 100 IF INKEY$="F" THEN LET A=1
 110 IF INKEY$="F" THEN PRINT AT
20,0;"                 "
 120 IF A=0 THEN GOTO 30
 130 LET H=H-1
 140 PRINT AT H,19;"▗";TAB 19;"▌
";TAB 19;"▟▖";TAB 19;"* ";TAB 19
;"  "
 150 IF X=39 AND H<4 AND H>0 THE
N GOTO 200
 160 IF H=0 THEN GOTO 1
 170 IF H=17 THEN GOTO 50
 180 GOTO 60
 200 CLS
 210 PRINT AT 3,19;"HIT"
 220 PAUSE 500
 230 RUN
 500 SAVE "COMET CRUSHER"
 510 RUN
```

Listing 2: Comet Crusher (8K ROM; 1K RAM).

```
   2 PRINT AT 20,19;"▒▒▒▒▒▒▒"
   4 LET X=0
   6 LET A=0
   8 LET I=2
  10 LET Y=30
  12 PLOT 40,2
  14 IF INKEY$="F" THEN LET A=1
  16 PLOT X,Y
  18 UNPLOT X,Y
  20 LET X=X+1
  25 LET Y=Y+RND*2-1
  50 IF NOT A THEN GOTO 14
  55 UNPLOT 40,I-2
  60 PLOT 40,I+1
  65 IF X=40 AND Y>I-2 AND Y<I+2
THEN GOTO 85
  70 LET I=I+1
  80 GOTO 16
  85 PRINT "HIT"
```





Crossing the Asteroid Belt

M. Hampson

You are the pilot of a space ferry going back and forth through the asteroid belt. This calls for highly skilled navigation to avoid hitting or being hit by the asteroids.

The asteroids are represented in your navigation tank by the O's, and the position of your ferry is indicated by the asterisk. You control your movement by the arrow keys 5 and 8.

If your ferry and an asteroid collide, the asterisk becomes inverse. Then you must get another ship. You build your piloting credentials by recording how many times you have successfully crossed and recrossed the belt. In 1K the computer cannot keep the count for you.

M. Hampson, 7 Hereford Dr., Clitheroe, Lancs BB7 1JP, U.K.

Reprinted with permission from *The Ultimate Magazine*.

Listing 1.

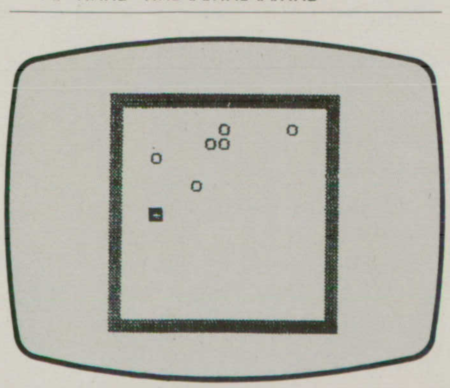
```
   1 REM 11111111111111111111111111
11111111122222222222222222222222
22222222233333333333333333333333
33333333344444444444444444444444
44444444455555555555555555555555
55555555566666666666666666666666
66666666677777777777777777777777
777777777
  10 FOR Z=16544 TO 16736
  20 INPUT X
  30 PRINT (STR$ (X+1000))(2 TO
);",";
  40 POKE Z,X
  50 NEXT Z
```

Listing 2.

```
006,016,197,014,000,205,245,008,
062,008,215,193,197,014,016,205,
245,008,062,008,215,193,016,234,
014,016,197,006,000,205,245,008,
062,008,215,193,197,006,016,205,
245,008,062,008,215,193,013,121,
254,255,032,230,033,130,064,126,
071,035,126,079,197,229,205,245,
008,062,000,215,225,193,035,126,
203,071,040,002,005,005,004,203,
079,040,002,013,013,012,043,043,
120,119,035,121,119,035,254,001,
032,002,203,142,254,015,032,002,
203,206,120,254,001,032,002,203,
134,254,015,032,002,203,198,229,
205,245,008,062,052,215,225,035,
062,160,189,194,215,064,058,033,
064,060,079,006,008,197,205,245,
008,042,014,064,126,254,052,193,
200,054,000,058,037,064,254,247,
032,004,013,032,001,012,254,239,
032,009,012,062,016,185,032,001,
013,000,000,121,061,050,033,064,
205,245,008,062,023,215,195,212,
064,
```

Listing 3.

```
   8 RAND
   9 FAST
  10 FOR Z=NOT PI TO VAL "27" ST
EP VAL "3"
  20 POKE VAL "16514"+Z,INT (RND
*VAL "3")+VAL "4"
  21 POKE VAL "16515"+Z,INT (RND
*VAL "8")+VAL "4"
  22 POKE VAL "16516"+Z,INT (RND
*VAL "4")
  30 NEXT Z
  40 POKE VAL "16417",NOT PI
  45 SLOW
  50 RAND USR VAL "16544"
  60 PRINT "■"
  70 RAND RND**RND**RND
```



The directions are as follows:

1) Do not use your RAM pack.

2) Enter the program in Listing 1. Be sure to type in line 1 as listed.

3) Type in RUN and ENTER.

4) Enter all the values in Listing 2. Be very careful. About halfway through the list, the ZX81 will run out of memory. Type CONT(inue) and continue entering the numbers.

5) Delete all the lines in Listing 1 except line 1 by entering the line number followed by ENTER.

6) Enter the lines in Figure 3. Note: the graphic in line 60 is an inverse asterisk (on B).

7) Type in RUN and ENTER and your journey across the asteroid belt begins.

DEF on the Sinclair

Jon Passler

Occasionally a program listing such as "3-D Plot" from Creative Computing's *Basic Computer Games* contains the DEFine statement which allows the programmer to define a function in the form

DEF FNA(X)=(expression)

FNA merely stands for FuNction A," and the variable X in the parentheses following FNA is termed the "dummy argument." Usually the expression to the right of the equals sign in the DEFine statement has at least one X in it. Of course, variable names other than A and X could be used.

A simple example would be the statement

DEF FNA(X)=X**2+X

Normally, when FNA is "called" or used within the program flow, an argument is given within the parentheses different from the dummy argument, but which replaces all occurrences of the dummy argument in the DEFine statement. FNA(2) would have the value of 2**2+2, or 6, while FNA(A+B) would be the result of (A+B)**2+A+B. FNA can be treated like any other numeric variable. The only difference is that it is the result of an expression DEFined at the start of the program, outside the program flow, and the programmer can alter the argument of the expression.

The definition could have contained more than one dummy argument or variables which are not dummy arguments, such as

FNB(F,G)=F*G-Q/R

Functions can also be used as arguments for other functions, for example

FNB(3,FNA(A+B))

would be the result of

3*((A+B)**2+A+B)-Q/R

The DEFine statement is useful to simplify equations or cut down on programming where one equation is used at several points within the program.

One way around DEFine in Sinclair Basic is to replace all FN calls with the expression in the definition. This often requires breaking down an equation to simplify it.

Another is to use the 8K VALue function which can evaluate a string such as "2" or "2+2" or even "X**2+X". The solution then is to replace DEF FNA with

LET A$=(expression)

and replace the FN call with VAL A$. The only problem appears when something like FNA(A+B) appears. We cannot call the function and define the argument all in one statement. Generally, if several different arguments are used in the FN calls, it is best to set the dummy argument equal to the argument before calling for VAL A$, setting, for example, X=A+B.

Listing 1 is the program for "3-D Plot" from *Basic Computer Games*. Here FNA(Z) is defined in line 100 as

30*EXP(-Z*Z/100)

and in line 150 the argument is

SQR(X*X+Y*Y)

which replaces all Z's in FNA. Line 150 also sets Z, which is not related to the Z

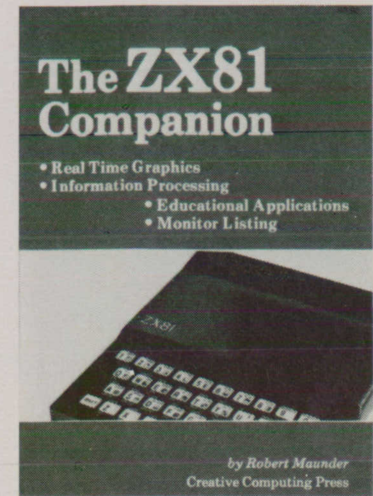
Listing 1: 3-D Plot, Original Program.

```
100 DEF FNA(Z)=30*EXP(-Z*Z/100)
110 FOR X=-30 TO 30 STEP 1.5
120 L=0
130 Y1=5*INT(SQR(900-X*X)/5)
140 FOR Y=Y1 TO -Y1 STEP -5
150  Z=INT(25+FNA(SQR(X*X+Y*Y))-.7*Y)
160 IF  Z<=L THEN 190
170 L=Z
180 PRINT TAB(Z);"*"
190 NEXT Y
200 PRINT
210 NEXT X
```

Jon Passler, 344 Cabot St., Beverly, MA 01915.

Get the most from your SINCLAIR

with these practical, program-filled books from *Sync*!

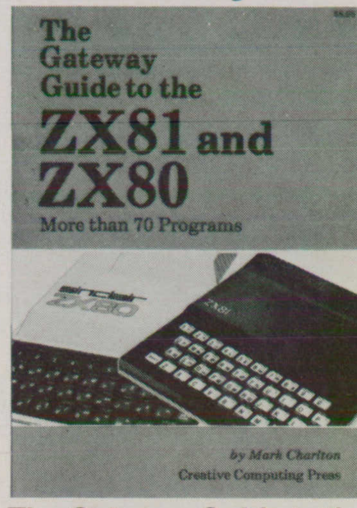


The ZX81 Companion
by Bob Maunder

The ZX81 Companion follows the same format as the very popular *ZX80 Companion,* and assists the ZX81 user in four applications areas: graphics, information retrieval, education and games. This practical guide contains scores of fully documented short routines plus complete programs and a disassembled listing of the ZX81 ROM Monitor. "Thoughtfully written, detailed, and illustrated with meaningful programs."—MUSE
5½" x 8", 132 pages. #17P $8.95 ($2.00)

Getting Acquainted With Your ZX81
by Tim Hartnell

This informative volume for the new ZX81 user contains more than 70 programs to help the reader get the most from his Sinclair computer. Game programs include Checkers, Alien Imploders, Blastermind, Moon Lander, Breakout, Star Burst and Derby Day. The book also shows programs for cascading sine waves, plotting graphs and tables, data sorting, equation solving, plus the use of PLOT, SCROLL, PRINT, TAB, PEEK, POKE and much more!
5½" x 8", 120 pages. #15Y $8.95 ($2.00)



The Gateway Guide to the ZX81 and ZX80
by Mark Charlton

The Gateway Guide is a practical programming manual for the beginner that furnishes over 70 fully documented programs. The majority of the programs have been written for easy conversion from machine to machine (ZX81, 4K ZX80 or 1K ZX80). *The Gateway Guide* describes each function and statement, illustrates it with a demonstration routine or program, and combines it with previously discussed material to help you understand your computer.
5½" x 8", 172 pages. #160 $8.95 ($2.00)

Computers for Kids (Sinclair Edition) by Sally Larsen

This new edition of *Computers for Kids* is written specifically to introduce children aged 8 to 13 to the ZX81. The book requires no previous knowledge of algebra, variables or computers, and it enables a youngster to program a ZX81 in less than an hour. There's also a section for parents and teachers. "*Computers for Kids* is the best material available for introducing students to their new computer."
—Donald T. Piele, Professor of Mathematics, University of Wisconsin-Parkside.
8½" x 11", 56 pages. #12S $3.95 ($1.00)

All volumes are softbound.

Creative Computing Press, **Dept. Z720, 39 East Hanover Ave., Morris Plains, NJ 07950**

Please send books listed below:

Item No.	Qty.	Price each	(P&H)	Total Price

Postage and handling charges appear in parentheses ( ) next to price of book.

Outside U.S.A. add $3.00 per order—shipped air mail only.

☐ Enclosed is $________ . NJ residents add 5% sales tax.
☐ Charge my (Charge and Phone Order $10 minimum):
☐ American Express ☐ MasterCard ☐ Visa

Card Number________________ Exp. Date________

Signature________________________

Mr./Mrs./Ms.________________________
(please print full name)

Address________________________

City/State/Zip________________________

CHARGE ORDERS—For Your Convenience PHONE
TOLL FREE 800-631-8112, In N.J. Only (201) 540-0445.

in FNA(Z), to an integer value using the FNA call. This program is translated into Sinclair Basic (8K ROM, 1K RAM) in Listing 2. Here the string variable A$ is used to replace DEFine and set the argument equal to

SQR(X*X+Y*Y)

in line 145 before calling for VAL A$. A$ is an input since several different functions are plotted. The same results could be obtained by adding

146 LET A=30*EXP(-Z*Z/100)

and changing VAL A$ to A in line 150, except that line 146 would have to be edited each time the function is changed.

The functions are slightly altered to run on either 1K or 16K RAM. Neither program, however, will work correctly with defective 8K ROMs. The plots appear within a circular X-Y plane tilted about 30 degrees toward the viewer with the curve rising above or falling below the plane.

Try the following functions with the 1K program:

20*EXP(-Z*Z/100)
20*SIN(Z/10)-15
SQR ABS(150-Z*Z)*.4-2
20*(COS(Z/16))-5
20-20*SIN(Z/18)

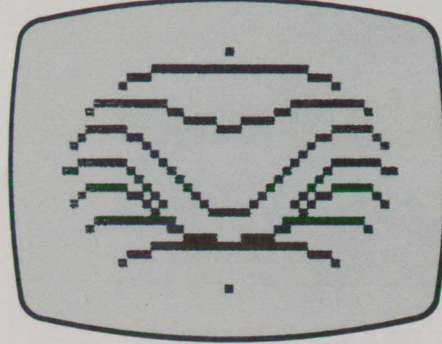
In the 16K version all the five functions above are held in F$ (see Listing 3). Each function word such as SIN uses only one byte in F$. At line 180 a zero is POKEd into DF-SZ, which is the system variable with the number of lines (usually two) in the lower part of the screen. Entering a zero into it allows printing the function on the 24th line. An input cursor will crash the system if called for when there is no room for it; so lines 330-340 replace a dummy input to stop the program temporarily (press any key to continue).

This is a rather unusual example of an occasion when using a string variable to hold a function or functions can be useful. The technique would also be useful where a function is used several times within a program to facilitate the translation of a DEFine statement or to highlight a function at the start of a program.

To use the programs, enter Listing 2, press RUN and NEWLINE and then enter one of the five functions above and observe the results. If you have 16K, enter the expanded version in Listing 3, press RUN and NEWLINE. However, since the functions are already included in the program, you do not have to enter them again. ■

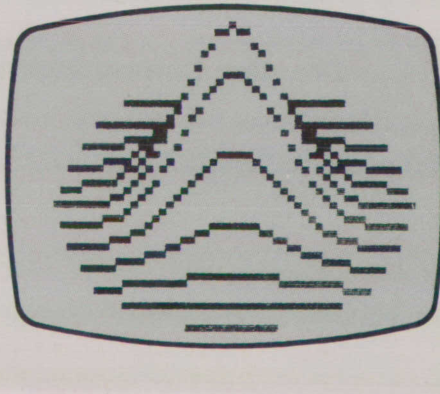
Listing 2: 3-D Plot for Sinclair (8K ROM, 1K RAM)

```
 100 INPUT A$
 110 FOR X=-20 TO 20
 120 LET L=0
 130 LET Y1=5*INT (SQR (400-X*X)
/5)
 140 FOR Y=Y1 TO -Y1 STEP -5
 145 LET Z=SQR (X*X+Y*Y)
 150 LET Z=INT (25+VAL A$-.7*Y)
 160 IF Z<=L THEN GOTO 190
 170 LET L=Z
 180 PLOT X+20,Z-5
 190 NEXT Y
 210 NEXT X
```



Listing 3: 3-D Plot for Sinclair (8K ROM, 16K RAM)

```
 100 REM 3-D PLOT
 110 DIM F$(5,16)
 120 LET F$(1)="25*EXP (-Z*Z/100
)"
 130 LET F$(2)="25*SIN (Z/10)-15
"
 140 LET F$(3)="25*(COS (Z/18))-
2"
 150 LET F$(4)="25-25*SIN (Z/18)
"
 160 LET F$(5)="25*EXP (-COS (Z/
16))-18"
 170 FOR B=1 TO 5
 180 POKE 16418,0
 190 PRINT AT 23,0;F$(B)
 200 PAUSE 99
 210 FOR X=-25 TO 25
 220 LET L=0
 230 LET Y1=4*INT (SQR (625-X*X)
/4)
 240 FOR Y=Y1 TO -Y1 STEP -4
 250 LET Z=SQR (X*X+Y*Y)
 260 LET Z=INT (25+VAL F$(B)-.7*
Y)
 270 IF Z<=L THEN GOTO 310
 280 LET L=Z
 290 PLOT X+30,Z-8
 300 PAUSE 10
 310 NEXT Y
 320 NEXT X
 330 PAUSE 9999
 340 IF INKEY$="" THEN GOTO 330
 350 CLS
 360 NEXT B
 370 STOP
```



GO TO LAS VEGAS FOR $1

Two Excellent LAS VEGAS Style Program - Listings For Only $1 (ZX81—16K)

Lucky 7 Slot Machine

* Styled after an actual "bell" 3 reel slot machine with correct payoffs and token distribution
* Fully animated - money goes into slot, handle is pulled, reels spin and stop in sequence, payoff coins fall into the payoff trough
* Lots of fun - a very addictive game

Vegas 21 Blackjack

* Most complete blackjack game available - split pairs, double down, ace's value automatically adjusted, dealer stands on 17, draws to 16
* Uses 5 decks, reshuffles when appropriate, will advise tens ratio and distribution of cards left
* A good way to learn the game or to practice before going to a real casino

Don't miss this opportunity to get these two excellent programs at such a low price. They are fully guaranteed - money back if not satisfied. Send $1 and a self addressed, stamped envelope to:

Florida Creations, P.O. Box 16422, Jacksonville, FL, 32216

ZX81 OWNERS with 16K RAM!
IT'S READY---ARE YOU?

SCOUT FIGHTER

$14.95 + 2.50 S&H

If so, you get:

*24-row/full-screen displays
*Nearly 2K of machine code for FAST real-time graphics
*A 4K Star Atlas as the moving backdrop during enemy engagements
*9-each fuel and skill levels that are interlocked for more "reality"
*4 ways to lose---no fuel, no oxygen, no gun power, or no shields. 1 way to win----destroy ALL enemy craft.
*An enemy data generator imbedded in the run as part of the "action"
*"Blink"-less pauses in BASIC
*Stick-on/peel-off custom keys
*In-depth booklet explaining both the "game" and the program
*Shipped on a certified cassette 1 copy each side

16K Programs for either ZX80-8K or ZX81

*PLANE FRAME-modelling for engineers
*FUNDAMENTAL ANALYSIS-for stocks investors
*POLSAT & GEOSAT-space age Ham Radio uses
*MATH PAK-Routines & Fit, Variance Analyzer, Statistics, and RPN Calculator
*E.S.P.-test your precognition
*Many others plus 6 1K 6-paks including Machine Code

ZETA SOFTWARE

WRITE for free catalog. (In Europe, send U$1 bill or DM2 stamp: DELTASOFT/Osterfeldstr.79D/D-2000 Hamburg 54 GER)

PO Box 3522
Greenville, SC 29608

Mail coupon or copy with payment to:
ZETA Software/PO Box 3522/Greenville,SC 29608

Enclosed is $______ ☐Check ☐__________
for ___ copy(ies) of SCOUT FIGHTER ZX81 16K
plus $2.50 S&H for the order.

NAME______________________________

ADDRESS___________________________

CITY/STATE/ZIP____________________
Please allow 4 weeks if paid by check.

Handling Strings from Another DIMension

Ken Berggren

Does this story sound familiar? One day, Fred found an excellent short program in a magazine article and he decided to translate it for his ZX80. But after entering only a few lines of the program, he ran into a statement like this:

250 DIM N$(4)

"What in the world does that mean?" he asked himself. When he could not answer that, he consulted a friend with a TRS-80. His friend explained that it works just like a regular DIM statement but with strings instead of numeric variables. That is, DIM N$(4) sets up four strings with the same name, N$, but individually numbered 1-4. Fred frantically flipped through the pages of the ZX80 manual and found the terse explanation of the DIM statement on page 89. But there was nothing there about strings. So, he set aside the magazine article and decided to stick with programs written just for the ZX80.

Well, if Fred sounds like some people you know, then consider this article dedicated to them. *SYNC* is a fine magazine, but it cannot publish every program that will work in the ZX80. And just because you cannot use a statement like DIM N$(4) does not mean that you cannot translate a program that uses it.

Now you 8K ROMers realize that your supercharged machines will DIMension strings without any hocus-POKEus, and you may be tempted to turn the page on me. But stick with me because I think that you will find this technique very interesting if not useful. Anyway, from now on we will be talking strictly in terms of the 4K ROM.

It is true that the people who designed the 4K ROM left out the ability to DIMension strings. However, with a few well-placed POKEs, you can plug up that hole in the 4K ROM.

Ken Berggren, 104 Ridgeway Ave., Louisville, KY 40207.

For example, enter this short program.

```
100 FOR I=1 TO 4
110 POKE 16450,I+37
120 INPUT A$
130 NEXT I
```

The A in line 120 is stored in the memory location 16450. The code for an A is 38, So, at present, a 38 is stored at the location 16450. When you RUN this program, it will execute lines 110 and 120 four times. The first time through, line 110 will POKE a 38 (1+37) into the location 16450. Then, line 120 will INPUT A$. But the second time through, line 110 will POKE a 39 (2+37) into location 16450. Since 39 is the code for a B, line 120 will then INPUT B$. This program actually changed itself! The third time through the loop, line 110 POKEs a 40 (3+37) into the memory location, and line 120 will INPUT C$. When the program is finished, it will have stored four strings: A$, B$, C$, and D$. You can imagine that these are numbered 1-4 because when I=1 you INPUT A$ and when I=4 you INPUT D$.

RUN the program and enter four words. Now change line 120 to 120 PRINT A$ and then GO TO 100. The method will work with INPUT, PRINT or any other string functions.

Some of you may thinking, "Big deal. What good is all this?" I think that the following two programs will illustrate the virtues of this technique.

The first program is a simple sort program. We all know that computers are very good at putting numbers in order. And since computers store letters as numbers, they are also good at putting words in order (alphabetical order, that is). The program will alphabetize up to 25 words and will display up to 22 of them.

Using the program is simple. Just enter the number of words you want to alphabetize and then enter the word with a NEWLINE after each one. When you have had the last word, the program takes over, and, a few seconds later, the words are displayed in alphabetical order.

I think that it is worth noting line 240. I do not know if other Basics let you use inequalities with strings, but ZX80 Basic does. That is a very nice feature. If you want to put the words in reverse order, simply reverse the inequality.

This program is not so great by itself. But it could be developed into a good utility program for handling a list of the names of friends for an address book or names of students for a grade book. It could possibly be adapted to help teach dictionary skills.

The second program illustrates the technique by computerizing a card game played something like Rack-O (by Milton Bradley). The POKEing is used to call each player by his name rather than the impersonal PLAYER 1, PLAYER 2, etc.

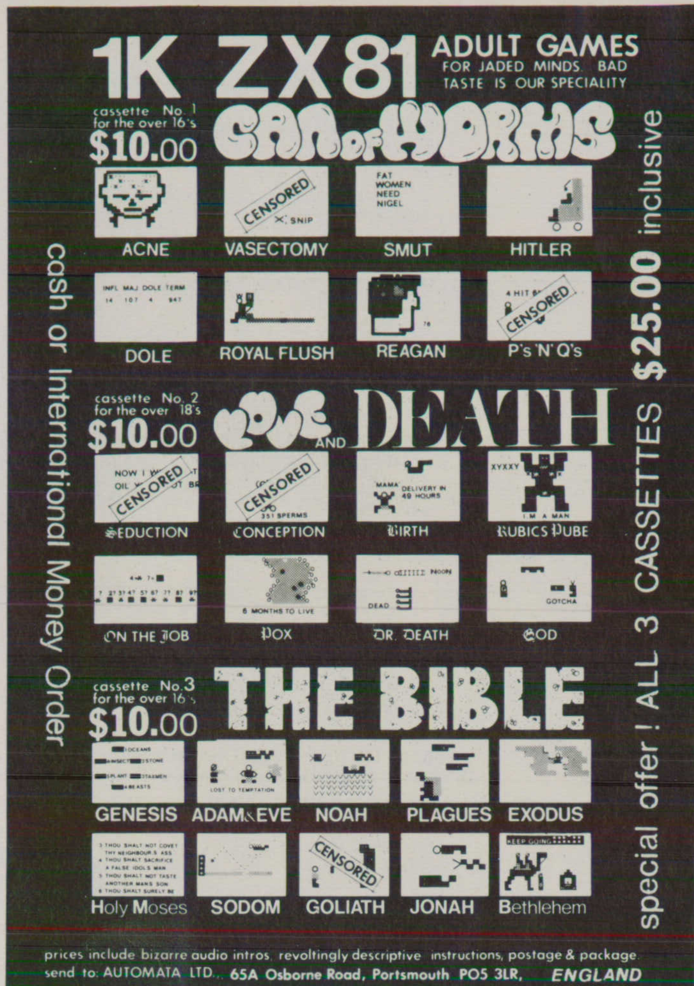
In this game, the players are dealt ten numbered cards. The remaining cards are placed face down, and the top card is turned face up to form a discard pile as in Gin. The object is to get ten cards in numerical order (not necessarily consecutive order). This is done by drawing a card from either pile and exchanging it for one of your cards.

In this version, of course, the computer handles all the cards. First, it shuffles them and places them into each player's "rack." Then the first player's cards are displayed and he is asked if he wants the card showing in the discard pile. If he does not, he enters "NO," and he is given a card from the face down pile. If he does not want that card, he enters "NO" again and his turn is over. If he decides to take either card, he enters "YES" and the computer will ask where in his "rack" he wants the card to go. The player then enters a number 0-9. His card will be placed in that position, and his turn will end.

At the end of each player's turn, the computer will display the cards that the player has in order so far. Then a NEWLINE will start the next player's turn.

The game ends when one player get his cards in order. To start a new game you have to RUN the program again.

If you have more than 1K of RAM, you could probably teach the computer how to play the game and then play against it. You could also allow more than four players. But be careful. Any modifications of these programs may affect the POKE locations. To make sure, LET I=0 and then GO TO the POKE statement in question. If the next statement does not contain a 9$ after you do that, you will have to change the POKE location until it does. But trial and error will not work very well if you make big changes or if you write your own programs. Then you will need a more exact method of finding the location of a specific byte in a program.



NEW RELEASE

A 44 KEY TACTILE FEEL KEYBOARD THAT PLUGS INTO ZX81 OR TIMEX SINCLAIR 1000. INCLUDES ALL GRAPHICS AND LEGENDS. 4 ADDITIONAL KEYS FOR EDIT, DELETE, QUOTE, FUNCTION. (NO SHIFTING REQUIRED). MEASURES 8½" x 4".

$70.00 MASS. RESIDENTS ADD 5% SALES TAX
SHIPPING & HANDLING $4.00

DELIVERY 6 - 10 WEEKS.
QUANTITY DISCOUNTS FOR USER GROUPS.
FOR MORE INFORMATION SEND SASE OR TO ORDER SEND CHECK OR MONEY ORDER TO:

E - Z KEY
SUITE 75
711 SOUTHERN ARTERY
QUINCY, MASS. 02169

STRUCTURAL DESIGN

For ZX - 81 8K/16K

Programs include GRAPHICS

STEEL BEAM DESIGN $29

12 Options of Beam Type and Loading

TIMBER BEAM DESIGN $29

3 Options of Beam Type and Loading

Choice of Timber Grade and allowable Stress

DORAN ENGINEERING, P.A.
113 South Chestnut Street
PLEASANTVILLE, NJ 08232
(609) 646-3111

VISA MASTERCARD

POCKET COMPUTER NEWSLETTER

A NEWSLETTER FOR POCKET COMPUTER USERS

This timely, compact publication provides up to the minute information on pocket computers, including models such as the Radio Shack TRS-80 PC-1 and PC-2, Sharp Electronic's PC-1500 and PC-1211, Casio, Panasonic/Quasar HHCs, and others as they are announced. We only cover PCs capable of executing a high level language such as BASIC.

☐ Up to the Minute News ☐ Product & Equipment Reviews
☐ Important Operating Tips ☐ Practical Programs ☐ More

By Subscription Only: for a calendar year period (January — December). You get all issues published to date for the calendar year in which you subscribe, at the time you subscribe.

MC/VISA Phone Subscriptions: (203) 888-1946

☐ 1981/82 Charter Subscriber (Issues 1 — 20). $40.00 in U.S. (U.S. $48.00 to Canada. U.S. $60.00 elsewhere.)
☐ 1982 Regular Subscriber (Issues 11 — 20). $30.00 in U.S. (U.S. $36.00 to Canada. U.S. $45.00 elsewhere.)
☐ Sample issue. $3.00 in U.S. (U.S. $4.00 elsewhere.) *Due to credit card minimum, this item cannot be charged.

Orders must be accompanied by payment in full. We do not issue invoices for the POCKET COMPUTER NEWSLETTER.
Thank you for your remittance.

Name: ________________________________
Addr: ________________________________
City: ________________ State: ___ Zip: ________
MC/VISA #: ________________ Expires: ________
Signature: ________________________________

POCKET COMPUTER NEWSLETTER
35 Old State Road, Oxford, CT 06483

One way is to count the bytes in each line and add them up. Every line has two bytes for the line number (no matter how many digits), one byte for the Newline at the end of the line, and one byte for every keystoke in between. So, in the example program, line 100 takes 9 bytes, line 110 takes 14 bytes, and line 120 takes 3 bytes up to the A for a total of 26 bytes. Add 26 to 16424, which is the starting address of every 4K ROM program, and you get 16450. That is the actual location of the A in memory. However, counting all the bytes in a long program is a very tedious chore. I found a better way.

When USR(24) is put into a program, it stops the execution of the program and returns the starting address of the next line. To illustrate, insert 115 PRINT USR(24) into the short program and then GO TO 115. 16458 appears on the screen because that is now the starting location of line 120. Line 115 takes 11 bytes so, when it is deleted, line 120 will move up 11 bytes to 16447. Adding the three bytes of line 120 again shows that the location of the A is 16450. USR(24) is an invaluable utility whenever you need to know the location of a byte in your program.

A lot of programs out there in books and magazines can be adapted to the ZX80. But sometimes you really have to work to get them to. Now the lack of DIMensioned strings is no longer a problem. There are other shortcomings in the 4K ROM in dealing with other Basics. Yet, with a little determination and ingenuity, you can usually get around them.

Program 1. Alphabetize (4K ROM; 1K RAM).

```
100 PRINT "HOW MANY ENTRIES?"
110 INPUT N
120 FORI=1 TO N
130 POKE 16478,I+37
140 INPUT A$
150 NEXT I

200 FOR I=1 TO N
210 FOR J=1 TO N-I
220 POKE 16537,J+37
230 POKE 16540,J+38
240 IF B$>C$ THEN GO SUB 900
250 NEXT J
260 NEXT I

300 FOR I=1 TO N
310 POKE 16584,I+37
320 PRINT D$
330 NEXT I
340 STOP

900 POKE 16658,J+37
910 POKE 16664,J+37
920 POKE 16667,J+38
930 POKE 16673,J+38
940 LET Z$=E$
950 LET F$=G$
960 LET H$=Z$
970 RETURN
```

Notes:

100-150: Get the words.
130: Changes A$.
200-260: Sort the words.
220: Changes B$.
230: Changes C$.
300-340: Display the words.
310: Changes D$.
900: Changes E$.
910: Changes F$.
920: Changes G$.
930: Changes H$.
900-970: Move the words.

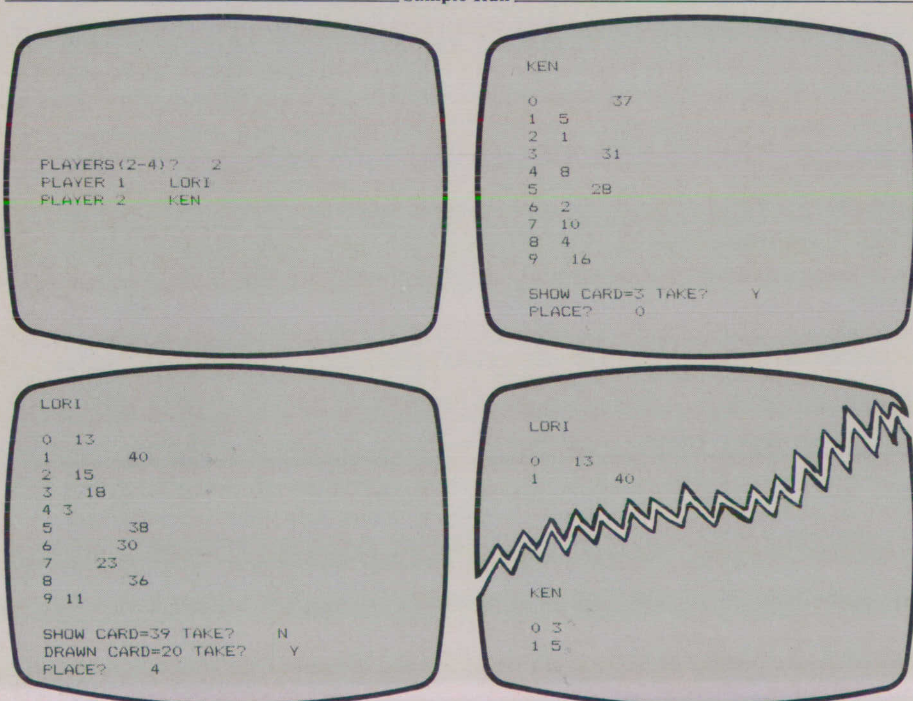
Program 2: Card Strings (4K ROM; 1K RAM)

```
100 PRINT "PLAYERS(2-4)?"
110 INPUT N
120 FOR I=1 TO N
130 PRINT "PLAYER#";I
140 POKE 16489,I+37
150 INPUT A$
160 NEXT I
190 LET E=20+10*N
200 DIM C(E)
210 FOR I=1 TO E
220 LET P=RND(E)
230 IF C(P)>0 THEN GO TO 220
240 LET C(P)=I
250 NEXT I
300 LET P=E-19
400 FOR I=1 TO N
410 CLS
415 LET T=0
420 POKE 16616,I+37
430 PRINT B$
435 PRINT
440 FOR J=0 TO 9
450 PRINT J;
452 FOR L=0 TO C(J*N+I)/N/4
454 PRINT "#";
456 NEXT L
458 PRINT C(J*N+I)
460 IF T=0 THEN GO TO 490
465 IF T>9 THEN GO TO 800
470 IF C(J*N+I)>C(J*N+N+I) THEN
    RETURN
480 LET T=T+1
490 NEXT J
495 PRINT
500 LET Y$="SHOW"
510 PRINT Y$;"#CARD=";C(P);"#TA
    KE?"
520 INPUT Z$
530 IF Z$>"X" THEN GO TO 600
540 IF Y$="DRAWN" THEN GO TO 65
    0
550 LET P=P+1
560 IF P>E THEN LET P=E-19
570 LET Y$="DRAWN"
580 GO TO 510
600 PRINT "PLACE?"
610 INPUT J
620 LET T=C(J*N+I)
630 LET C(J*N+I)=C(P)
640 LET C(P)=T
650 CLS
660 LET T=1
670 GO SUB 430
680 INPUT Z$
700 NEXT I
710 GO TO 400
800 POKE 16988,I+37
810 PRINT C$;"#WINS"
```

Notes:

100-160: Get the players.
190: E=total number of cards.
200-250: Shuffle the cards.
300: P=pointer to show card.
410-490: Display a player's cards.
452-458: Spaces each card over by magnitude.
465-480: Part of winner test subroutine.
500-580: Players pick their cards.
600-640: Chosen card put in "rack."
650-680: Test for winner.
Display cards in order so far.
700: Next player's turn.
710: Back to player 1.
800-810: Print the winner.

Sample Run



8K ROM Versions

Although the article is intended to help 4K ROM users, we thought the 8K ROM users might like to use the programs so the 8K ROM Versions are also given below in Programs 3 and 4.

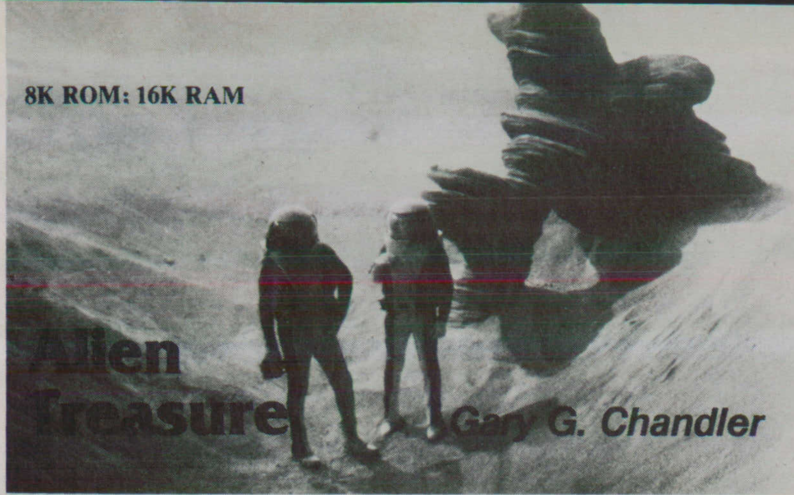
Program 3: Alphabetize (8K ROM; 1K RAM).

```
100 PRINT "HOW MANY ENTRIES?"
110 INPUT N
115 DIM A$(N,6)
120 FOR I=1 TO N
140 INPUT A$(I)
150 NEXT I
200 FOR I=1 TO N
210 FOR J=1 TO N-I
240 IF A$(J)>A$(J+1) THEN GOSUB
900
250 NEXT J
260 NEXT I
300 FOR I=1 TO N
320 PRINT A$(I)
330 NEXT I
340 STOP
940 LET Z$=A$(J)
950 LET A$(J)=A$(J+1)
960 LET A$(J+1)=Z$
970 RETURN
```

Program 4:
Card Strings (8K ROM; over 1K RAM).

```
100 PRINT "PLAYERS(2-4)?"
110 INPUT N
115 DIM A$(N,10)
120 FOR I=1 TO N
130 PRINT "PLAYER ";I
150 INPUT A$(I)
160 NEXT I
190 LET E=20+10*N
200 DIM C(E)
210 FOR I=1 TO E
220 LET P=INT (RND*E)+1
230 IF C(P)>0 THEN GOTO 220
240 LET C(P)=I
250 NEXT I
300 LET P=E-19
400 FOR I=1 TO N
410 CLS
415 LET T=0
430 PRINT A$(I)
435 PRINT
440 FOR J=0 TO 9
450 PRINT J;
452 FOR L=0 TO C(J*N+I)/N/4
454 PRINT " ";
456 NEXT L
458 PRINT C(J*N+I)
460 IF T=0 THEN GOTO 490
465 IF T>9 THEN GOTO 800
470 IF C(J*N+I)>C(J*N+N+I) THEN
RETURN
480 LET T=T+1
490 NEXT J
495 PRINT
500 LET Y$="SHOW"
510 PRINT Y$;" CARD=";C(P);" TA
KE?"
520 INPUT Z$
530 IF Z$>"X" THEN GOTO 600
540 IF Y$="DRAWN" THEN GOTO 650
550 LET P=P+1
560 IF P>E THEN LET P=E-19
570 LET Y$="DRAWN"
580 GOTO 510
600 PRINT "PLACE?"
610 INPUT J
620 LET T=C(J*N+I)
630 LET C(J*N+I)=C(P)
640 LET C(P)=T
650 CLS
660 LET T=1
670 GOSUB 430
680 INPUT Z$
700 NEXT I
710 GOTO 400
810 PRINT A$(I);" WINS"
```

8K ROM: 16K RAM



Bruce Birmelin/Paramount Pictures.

Alien Treasure

Gary G. Chandler

The 8K ROM and 16K RAM have given ZX80/81 users new possibilities in programming their own games. *Alien Treasure* is an example of how some of these new capabilities may be used.

In *Alien Treasure* two kinds of treasure are scattered over the surface of a planet far from earth. Your ship has landed, and you have sent out your robot searcher which you guide from your control room. The field of search shows on your ZX81 as a full screen display. The treasure is marked by inverse video periods which are worth one point each and by inverse video asterisks which are worth 10 points each. (You can set the values in your own currency if you wish.) However, there are always hazards in searching for treasure. In this case a powerful monster is guarding the search area. While your robot is attempting to pick up the treasures, you must guide it to avoid this monster which appears on your screen as an inverse video 0. Your robot, shown as a graphic square, is moved about by the use of the cursor control keys 5, 6, 7, and 8. The robot also can take a jump away to a random position in the same column by using the 0 key. However, you must use this 0 key with caution because the robot may land right on top of the monster.

The current total of your successful treasure gathering is displayed on the screen in the upper left corner. The total you must beat is displayed to the right of yours.

The game uses a 22 * 32 array to keep track of the treasure locations. When the game is run, it takes about 20 seconds for the screen and the array to be set up and loaded with the proper values. So do not hit the BREAK key too soon because you think the program is in an infinite loop.

Before you SAVE this program, execute the CLEAR command. Otherwise you will save the entire array along with the program, and it will take much longer to SAVE and LOAD in the future.

Alien Treasure has proved to be an interesting and challenging game for those who have tried it so get out your keyboard and start the search.

Line notes:

30: inverse space
36: inverse asterisk
44: inverse period
65: inverse zero
66: inverse graphics on A
73: inverse letters in ""
85: inverse space
89: inverse space

Gary G. Chandler, ATU Box 283, Russellville, AR 72801.

```
   1 LET C1=0
   2 REM TREASURE CHASE
   4 REM BY GARY G. CHANDLER
   5 LET X=0
   6 LET Y=16
   7 LET Y1=Y
   8 LET X1=X
   9 LET A=INT (RND*22)
  10 LET B=INT (RND*32)
  12 LET C=0
  20 DIM H(22,32)
  26 FOR I=0 TO 21
  28 FOR J=0 TO 31
  30 PRINT AT I,J;"█"
  31 IF I=0 OR J=0 THEN GOTO 50
  32 IF NOT RND<.05 THEN GOTO 40
  34 LET H(I+1,J+1)=2
  36 PRINT AT I,J;"█"
  40 IF NOT RND<.1 THEN GOTO 50
  42 LET H(I+1,J+1)=1
  44 PRINT AT I;J;"█"
  50 NEXT J
  60 NEXT I
  65 PRINT AT X,Y;"█"
  66 PRINT AT A,B;"█"
  67 IF H(A+1,B+1)=2 THEN LET C=
C+10
  68 IF H(A+1,B+1)=2 THEN LET H(
A+1,B+1)=0
  69 IF H(A+1,B+1)=1 THEN LET C=
C+1
  70 IF H(A+1,B+1)=1 THEN LET H(
A+1,B+1)=0
  72 IF C>C1 THEN LET C1=C
  73 PRINT AT 0,0;"█████";C;AT 0,
9;"██████████";AT 0,21;C1
  80 PAUSE 9000
  81 IF A>X THEN LET X1=X+1
  82 IF A<X THEN LET X1=X-1
  83 IF B>Y THEN LET Y1=Y+1
  84 IF B<Y THEN LET Y1=Y-1
  85 PRINT AT X,Y;"█"
  86 LET X=X1
  87 LET Y=Y1
  88 IF X=A AND Y=B THEN GOTO 20
0
  89 IF NOT INKEY$="" THEN PRINT
 AT A,B;"█"
  95 IF INKEY$="0" THEN LET A=IN
T (RND*22)
  98 IF INKEY$="5" THEN LET B=B-
1
 100 IF INKEY$="8" THEN LET B=B+
1
 110 IF INKEY$="6" THEN LET A=A+
1
 120 IF INKEY$="7" THEN LET A=A-
1
 132 IF A>21 THEN LET A=21
 133 IF A<1 THEN LET A=1
 134 IF B>31 THEN LET B=31
 135 IF B<0 THEN LET B=0
 140 IF X=A AND Y=B THEN GOTO 20
0
 160 GOTO 65
 200 PRINT AT 10,10;"**GOT YOU**
"
 202 PRINT AT A,B;"X"
 205 PRINT AT 20,2;" PLAY AGAIN
INPUT Y OR N "
 207 INPUT G$
 208 IF G$="N" THEN STOP
 210 GOTO 5
```

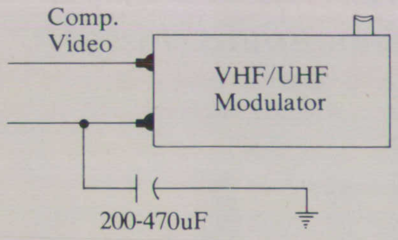
*Ed.—***A WORD OF CAUTION:** *Any hardware project for your computer must be approached with extreme caution.* **SYNC** *cannot be responsible for any problem that may arise from attempting hardware projects. Obviously, any damage to your computer can be costly in time and money.*

hardware tips

Thick Black Bars

George R. Ingle

In *SYNC* 1:6 Cecil Bridges points out that the thick black bars on the display may be caused by a failing capacitor in the power supply. However, it is not necessary to crack open the mains power adapter to replace the 1000uF capacitor. A simpler solution is to add another capacitor, rated 200-470uF at 25wv, across the +5v and ground connection to the UHF/VHF modulator. This not only corrects the leaking capacitor problem but also greatly reduces the "lining" of the display caused by the ZX80 display circuit itself. Obviously, a miniature capacitor should be used because of the minimal free space inside the computer case.



Insufficient Filtering Solved

Robert D. Hartung

Like many others I found that the DC filtering capacitance of the power supply included with the 16K RAM pack is borderline in supporting the demands of both the RAM pack and the ZX80 with its 17 extra ICs as compared with the ZX81. The symptoms of insufficient filtering include false LOADs from the excessive 60 Hz ripple in the DC and a horizontal line or blank bar moving vertically through the TV display every two seconds. If these symptoms disappear when the RAM pack is disconnected, either some component is drawing excessive current or, more likely, the added load of the RAM pack is causing inadequate filtering of the unregulated DC power to the computer.

Adding a 2200uF 50VDC capacitor across the output wires near the power jack eliminates the problem. Use a VOM or an LED tester to determine which is the positive output wire and which is the negative in order to be sure that you are observing the proper polarity in connecting the capacitor. This capacitor also gives some protection against momentary drop-outs occurring on the main power lines.

A note of caution: since a capacitor of this size stores considerable energy even after the power pack is unplugged from the AC outlet, I strongly recommend inserting a small lever type microswitch (e.g., Radio Shack 275-016) in one of the DC wires near the computer power plug. This will avoid possible burning or fusing of the power jack outlets when the plug is inserted or withdrawn when the capacitor is charged. The switch gives a bonus of being a "panic" switch to get out of endless program sequences as well as to cut the power off when connecting or disconnecting the 16K RAM pack.

To install the switch, cut and strip one of the power cord wires as near to the computer plug as practicable. Solder one wire end to the C lug of the switch and the other to the NC lug (normally closed). An LED pilot light can be added at the same time. Strip, but do not cut, one-half inch of the other power cord wire. Again, determine the polarity of this wire with the wire which goes from the switch lug to the computer plug. Solder the LED lead which is nearest the flatted side of the LED base (cathode) to the negative wire. Solder a 1K resistor to the other LED lead and in series to the wire which is positive. Carefully tape all bare leads and connections to isolate them from each other. Since only 45 grams of pressure will open the switch, taping it to the power cord and plug will give all the necessary support.

Robert D. Hartung, PO Box 125, Palmyra, NY 14522.

Memory Expansion Power Supply

George R. Ingle

David Sommers mentions (*SYNC* 1:6) the problem of using an additional power supply with his memory expansion. For those building an additional memory expansion unit, the following suggestions might be helpful.

1) Insure that all components are using a common ground return.

2) Do not connect the +5v regulated voltages or +9 to +12 unregulated voltage in parallel. Instead, use a common ground whether earth or floating, depending on the design, and feed the additional memory unit with a separate +5v regulated supply.

3) Insure that the additional power supply is well-filtered and uses, if possible, a three prong grounded outlet and AC supply cord.

(Ed.—A schematic of the author's power supply that provides for +15v reg., +12v reg., +5v reg., and -12v unreg. is available from the author for $1.00 and a SASE.)

George R. Ingle, 24593 1/2 Monterey Ave., San Bernadino, CA 92410.

USE OUR PLANS AND SAVE $$$$$

SYNCHRONIZE YOUR SINCLAIR

KEYBOARD how to add a full-size keyboard to a ZX/80 or 81. Includes instructions on how to use any extra keys for Shift Lock, Automatic shift, Automatic Repeat, and Reset. $5.00

POWER SUPPLY Running a little hot? Design and build your own supply to fit your needs at a fraction of wired price. Includes how to estimate both present and future needs and covers both single and multi-voltage. $5.00

BUSS Inexpensive exspansion system unlimits your edge connector. $3.50

JOYSTICK Add popular Atari joysticks and find out how much fun games can be. $2.50

Send check or money order to:

SYNCHRONIZE
Box 1667
Kerrville, Texas 78028

PRACTICAL ZX-81TM SOFTWARE

ZX-PANDING LTD

ZX-81TM Not Just For Games Anymore!

ZX-PANDING, LTD., an American company, uses the tremendous ZX-81Tm data handling capabilities to make our daily tasks easier. Programs are on cassette, thoroughly tested, with easy-to-read printed instructions, written in BASIC to allow tailoring to your special needs. Send a STAMPED SELF-ADDRESSED ENVELOPE for details on these and other programs.

1. YOUR SPECIAL DAY--1K--Good example of using successive equations to answer a practical question. Enter any date in history, and ZX-81Tm will tell you the day of the week on which it occurred. Useful in many fields and a great ZX-81Tm demonstration. . ONLY $3.75*
2. CLOCKS AND TIMERS--1K--A digital clock (standard or military time), a count-up timer (for telephone calls), and a count-down timer (useful in cooking). Another great ZX-81Tm demonstration. . . . . ONLY $3.75*
3. ANYPOINT PLOTTER--1K--Crunches any data to allow graphing any positive data points: stock prices, temperature, experimental data, etc. . . . ONLY $3.75*

ALL 3 FOR ONLY $8.75!

4. CHECKBOOK, INCOME TAX AND BUDGET ORGANIZER--16K A powerful yet easy to use finance program--10 jobs in menu driven format. Checks and deposits are totaled and itemized under desired categories for budget planning and income tax preparation. Makes checkbook balancing easy. Ninety transactions manipulated at one time with 16K. . . . ONLY $13.75*

ZX-PANDING, LTD.
P.O. BOX 25
NEWTON, NC 28658
704-464-2742

*ALL orders please add $1.25 to total order to cover POSTAGE AND HANDLING. (Foreign orders add $2.00 to cover air costs. Your payment must be in a U. S. dollar draft payable to U. S. bank.)

Sinclair ® Computer Owners

YOU NEED THIS!

- Protects against power line glitches
- Provides extra power for add-ons
- Cuts heat in the internal regulator

The HL-1205 uses the same modern switching regulator technology as used on big computers to trade harmful heat for extra power. so ... Just plug your ZX81 A.C. Adapter into the HL-1205, plug the HL-1205 into the computer and turn on the **switch**. Add a rechargable 12 volt battery (not supplied) to have an **UNINTERRUPTABLE POWER SUPPLY**, or operate from 12V car or truck system. **A MUST FOR SERIOUS COMPUTER USERS**

$35.95 plus $4.00 shipping and handling

30 day money back guarantee

To order send check to:
Hurricane Electronics Lab
P.O. BOX 1280
HURRICANE, UTAH 84737
[801]635-2003

MUSIC and KALEIDOSCOPES

VIRTUOSO. A music synthesis program for the Timex/Sinclair and ZX81 with 2K or more. NO MUSIC KNOWLEDGE NECESSARY! 96 notes. range from put put to above treble clef, any tempo. Whole to 32nd notes. Songs may be saved for future use. Required hardware? - - Your cassette recorder. $6.95 pp.

PLAYER ZX81. A less versatile music program for 1K memory. Music instruction not included. $6.95 pp.

SQUARES and CROSSES, Kaleidoscopic simulations for the same machines. 1K or more. $6.95 pp.

All on standard cassette. Guaranteed to load. Music programs include random sound listing or save copy. Add $3.00 each, outside U.S.

Mail order only.:
William Don Maples
688 Moore St.
Lakewood, CO 80215.

Ear Input Circuit

James Dowell

I upgraded my ZX80 (UHF) with the 8K ROM and MicroAce Video Upgrade kit. I had no loading problems while I was using a Sears tape recorder. Later I procured a Craig tape recorder as a dedicated component of my system. Then my loading problems began with no apparent permanent solution.

Upon inspection of a defunct ZX80 (VHF) with supposed loading problems, I discovered that Sinclair had made two minor modifications to the ear input circuit. One was the addition of a .01mf capacitor across the terminals of the input jack (RFI filter). Removal of this capacitor restored the loading capability of this unit. The other modification was the addition of a 6.8K resistor from the connecting bus between C12 and R1 to the +5V bus. Installing a similar resistor in my original ZX80 cleared all of my loading problems and greatly increased the dynamic range of the input circuit.

Top Line Hook Solution

James Dowell

Tom Keeney's fine article (*SYNC* 2:1) points to a problem with the MicroAce Video Upgrade kit and the top line. A solution to this "top line hook" in SLOW mode is as follows:

Connect a capacitor (.027-.030mf) from pins 10-11 of IC6 to ground. Using the space of the unused C9, insulate the "hot" lead of the capacitor and pass it through the +5V hole for C9 (next to pin 14 of IC6) and connect to pins 10-11 of IC6. Connect the ground lead of the capacitor to the ground hole for C9. Too little capacitance will not quite correct the hook whereas too much capacitance will over-correct the hook.

I made a large cutout in the top cover of the ZX80 and cemented a 4x6x3/4 inch plastic box over the opening. This gave me room to mount the Video Upgrade, Keyboard Beeper, Video Reverse Switch, and a Reset Switch in the ZX80.

Although my ZX80 with Video Upgrade works fine with a 'defective' 8K ROM, I have been unable to make it work with the replacement ROM (which works fine in a non-upgraded ZX80).

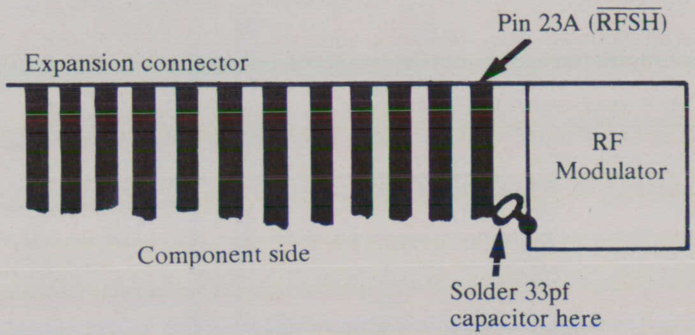
James Dowell, 735 Myra Ave., Chula Vista, CA 92010.

Problems in ROM Changing

Herb Hornung

A common problem which occurs when replacing the 4K ROM with the 8K ROM is that the computer does not turn on every time (or even at all) when it is plugged in. If you have this problem, you can solve it by soldering a 33pf capacitor from REFRESH to ground. That is, solder the capacitor from pin 23A of the expansion connector to the ground connection of the RF modulator (case). Keep the leads as short as possible (see Figure 1). After performing this modification, I checked to see if the 16K RAM pack still worked and it did.

Figure 1.



Strong Signals on KBD 0 through KBD 4

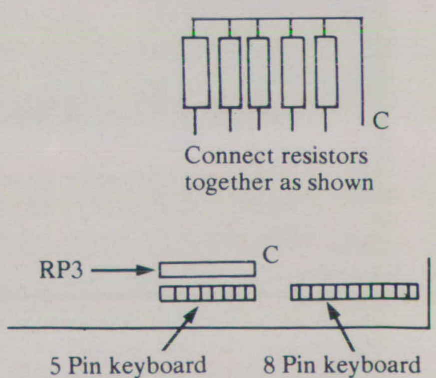
Herb Hornung

Quite a few ZX81s that I have seen have very strong signals on KBD 0 through KBD 4. This can cause the following problems: 1) some shifted functions will not work; 2) some characters are always shifted; 3) some characters will not print at all. These problems may appear all the time or only after adding a printer, plugging in a 16K memory module, or upgrading the keyboard.

This condition can be corrected by removing the 10K Ω resistor pack (RP3) and replacing it with 8.2K Ω resistors. Solder one lead from each resistor to the KBD 0 through KBD 4 a solder the other leads together and to the "C" contact of the circuit board (see Figure 2). In some extreme cases it may be necessary to change the resistor to as low as 6.8K Ω.

Ed.—Herb Hornung is interested in hearing from readers who have hardware problems or information. He will attempt to help (no charge) if a stamped, self-addressed envelope is enclosed.

Figure 2.



Herb Hornung, Double H Electronics, 195 Lelani, San Antonio, TX 78242.

resources

The "Resources" column lists new products for Sinclair users. Suppliers and users are invited to send brief product descriptions and ordering details to: Resources, SYNC, 39 E. Hanover Ave., Morris Plains, NJ 07950.

User Groups

- North Alabama ZX80/1 Users Group. For details contact:
  Bob Boyer
  1103 Rivlin Rd.
  Huntsville, AL
  (205) 883-4354 (evenings)

- San Francisco Bay Area Sinclair ZX Users Group (ZUG). Publishes newsletter *SincLink*. For details contact:
  Paul D. Perreault
  c/o Stanford Telecommunications, Inc.
  1195 Bordeaux Dr.
  Sunnyvale, CA 94086
  (418) 734-5300, x267

- Westinghouse ZX80/1 Users Club. Newsletter. $1 contribution appreciated. For details contact:
  Jack Fogarty
  Westinghouse MS 3525
  PO Box 1521
  Baltimore, MD 21203

- Pittsburgh Area Computer Club (Special Interest Group: Sinclair). For details contact:
  Dick Walsh
  1605 Middlecrest Dr.
  Glenshaw, PA 15116
  (412) 487-0789

- Chattanooga Area Sinclair Users. For details contact:
  Dan Williams
  PO Box 1321
  Collegedale, TN 37315

Forming a User Group

- Any Evanston, IL, area users interested in forming a group? Contact:
  Brendan P. Holly
  1246 Elmwood Ave.
  Evanston, IL

ROM Disassembly

- *Sinclair ZX81 ROM Disassembly. Part A* (the operating system) by Dr. Ian Logan, $15.00. *Sinclair ZX81 ROM Disassembly, Part B* (calculator routines), $17.00. Sent direct by airmail by the author. U.S. personal checks accepted. Spectrum books in development.
  Logan Software (Lincoln)
  24 Nurses Lane, Skellingthorpe
  Lincoln LN6 0TT
  United Kingdom

- *The ZX80 1K Disassembler* for the 4K ROM ZX80 and *The ZX81 1K Disassembler* for the ZX81 and 8K ROM ZX80. Begins with starting address, a key is calculated and displayed, manual then gives mnemonic, all numbers displayed in decimal, about 100 bytes of RAM to store MC program for disassembly; RAM memory test provided; addresses of bytes failing the test are displayed. Manual, reference cards, and cassette with disassembler and memory test programs. $9.95 pp. for either 4K or 8K. Further information and catalog upon request.
  Lamo-Lem Laboratories
  Box 2382
  La Jolla, CA 92038-2382

- *ZXBUG V3.125*. Machine code monitor/debugger with full ZILOG disassembler. Just under 4K; resides at top of the 16K ZX81; 30 commands. Allows: search of any block for any occurrence of a byte or word, display and altering of main and auxiliary registers, examination of flags, replacement in a given block of all occurrences of a byte with another, moving blocks around in memory, display of Z80 ZILOG Mnemonic disassembler page by page. £7.00.
  Artic Computing Ltd.
  396 James Reckitt Ave.
  Hull, North Humberside
  United Kingdom

Catalogs/Directories/Indexes/

- *Software Vendor Directory*. (6th ed.). Cross referenced index covering 22 systems and over 12,000 software products in 300 categories obtainable from 1,800 software vendors and 123 hardware vendors. $57.95; with 2 updates in 12 months: $100 or $260 on disk. Contact:
  Micro-Software Services, Inc.
  PO Box 482
  Nyack,NY 10960
  (914) 358-1340

- ZX80/81 Users' Library. A catalog in 3-ring binder format with programming tips and abstracts of programs submitted by subscribers and checked by library staff for accuracy and usefulness. Periodic newsletter. Program prices: $1.50 for 1K, $3.00 for 16K. Annual subscription: $25 for U.S.; $32 for non-U.S. For further details contact:
  ZX80/81 Users' Library
  Suite 434
  4614 Kilauea Ave.
  Honolulu, HI 96816

Periodicals

- Imprint Software review program. A quarterly newsletter of software reviews by category. For details contact:
  Software Review
  Imprint Software
  420 South Howes
  Ft. Collins, CO 80521
  (303) 482-5574; Telex 45-4590
  In U.K.:
  Software Review
  Imprint Software
  16 Milton Ave.
  Highgate, London N6

- *ETC: Educational Technology and Communication.* A monthly newsletter to help schools, esp. decision makers, to move into the new technology. $36 for one year; $60 for two years.
  ETC Subscription Department
  Far West Laboratory
  1855 Folsom St.
  San Francisco, CA 94103

Programming Programs

- *PEEK* displays 100 memory locations in decimal (Basic); *BASE* gives conversions of octal, decimal, binary, hex, hibyte/lobyte (Basic); *PPU* a PEEK and POKE Utility, a fully documented monitor (Basic and mc). All four programs on standard cassette and full documentation (64 pp.; 8 1/2 x 11) for $24.17. *Bidirectional Dictionary* gives 2 lists: from Decimal OP CODE arranged numerically to Assembler Mnemonic and from Assembler Mnemonic arranged alpha-numerically to Decimal OP CODE.
  Joseph L. Hartmann
  101 Tonnele Ave.
  Jersey City, NJ 07306

Educational Programs

- "Fun to Learn" series. 8 cassettes: English Literature I and II, Geography, History, Mathematics, Inventions, and Music. $12.95 per cassette. $1.95 s&h per order.
  Sinclair Research Ltd.
  3 Sinclair Plaza
  Nashua, NH 03061

Business/Household

- *Vu-Calc* constructs large tables for purposes such as budgets, finances, and projections ($14.95). *Vu-File* storage and retrieval for collections, accounts, directories, membership lists ($14.95). *The Collector's Pack* holds up to 400 records in each of 6 categories ($17.95). *The Club Record Controller* holds names, addresses, phone numbers, and 5 additional categories for up to 100 members on one cassette ($17.95). $1.95 s&h per order.
  Sinclair Research Ltd.
  3 Sinlair Plaza
  Nashua, NH 03061

- *ZX81 Personal Banking System.* Keep detailed records of your finances; uses unique machine code to store all transactions in datafiles on cassette separate from the program. Cassette and 12 page manual for $20. Large SAE for details.
  J. P. Gibbons A.I.B.
  14 Avalon Road
  Orpington, Kent, BR6 9AX
  United Kingdom

- *Check Register, Inventory,* and more programs for the ZX81 (16K to 64K). Makes full use of the 64K. Can be user defined to your RAM size. On high energy cassette $9.95 plus $1.00 s&h.
  Brittanysoft, Ltd.
  PO Box 1043
  Twin Falls, ID 83301
  (208) 324-3086

- *Coin Collection, Stamp Collection,* and *Baseball Card Collection.* Database inventory programs using menu and descriptive prompts for ZX81 and ZX80 or MicroAce with 8K ROM. 16K RAM. Cassette (guaranteed loading) and manual. Each title $4.95 plus $1.05 s&h. SASE for free catalog.
  M. C. Hoffman
  PO Box 117
  Oakland, NJ 07436

Graphics Programs

- *GRAPHIX.* Image processor. No more tedious number punching to "draw" on the screen. Features: transferable images, image mixer, image reverser, up to 12 "files." ZX81/ZX80 (8K ROM); 16K RAM. $5.99 incl. postage.
  Nick Godwin
  4 Hurkur Crescent, Eyemouth,
  Berwickshire TD14 5AP
  Scotland

- *The Artist* creates drawings and paintings on the screen. Features: user's choice of brushes and backgrounds, free mixing of graphics and alphanumerics, easy specification of lines and circles, ability to store complex patterns and reproduce them anywhere on the screen. Cassette and instructions for $10.
  KSOFT
  845 Wellner Rd.
  Naperville, IL 60540

RAM Expansion

- EconoTech 16K RAM Pack. Uses NMOS dynamic RAMs, standard 4116 memory chips, and Sinclair power pack. Tight-fitting 44-way edge connector with gold-plated contacts plugs into expansion port; designed to prevent wobble. Compatible with ZX Printer. RAM pack with 6-month guarantee and comprehensive instructions £19.95 (including VAT) plus £1.50 s&h to anywhere in the world. For further details contact:
  EconoTech
  30 Brockenhurst Way
  London SW16 4UD
  United Kingdom

- 2K RAM kit for ZX81 users with 1K RAM. Increases program size (with filled screen) by more than 10 times. Kit includes: 2K-byte IC, 2 part socket, jumper, solder, and easy instructions. $29.95 postpaid.
  Micro Logic Corp.
  PO Box 174-ZB
  Hackensack, NJ 07602
  (201) 342-6518

Light Pen

- Light pen for ZX81 users. Palm-sized plug-on module with three modes: DRAW, ERASE, and CLEAR. Allows entering complex graphics by just touching the screen. Upper 90% of screen is free for graphics; lower portion divided into three zones for changing modes. Virtually eliminates the PRINT and PLOT statements. Images held in strings to be stored on cassette if desired. $69.95.
  ZODEX
  East Hill
  Oakham, MA 02068

Interference Control

- Toll Free Interference Control Hot Line. Experienced staff will analyze problem situations and make specific recommendations for control of processor or peripheral interference. Hot line: 1-800-225-4876 between 9 a.m. and 4 p.m. Eastern Time Monday through Friday. Free 40 page Interference Control Product Catalog.
  Electronic Specialists, Inc.
  171 S. Main St.
  PO Box 389
  Natick, MA 01760
  (617) 655-1532

Fairs and Workshops

- MICROSCENE BRUM 82. A big ZX80/ZX81/Spectrum show in the center of Britain's second city: Bingley Hall, Birmingham 1 on 11 September, 1982. Advance tickets £1; advance brochures £1. Send to:
  Microscene
  6 Battenhall Road
  Harborne
  Birmingham B17 9UD
  United Kingdom

Games

- *Backgammon* ($9.95), *Chess* ($12.95), *Fantasy Games* ($8.95), *Space Raiders and Bombers* ($8.95), *Flight Simulation* ($9.95), 8 different *Super Program* cassettes ($8.95 ea.). $1.95 s&h per order.
  Sinclair Research Ltd.
  3 Sinclair Plaza
  Nashua, NH 03061

Index to Advertisers

EZRA GROUP II

EZRA GROUP II